STRESS ANALYSIS OF CERAMIC GAS TURBINE BLADES BY THE FINITE ELEMENT METHOD PART II.

John H. Preisel, Jr.

Report Number: NPS69-78-010

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

STRESS ANALYSIS OF CERAMIC GAS TURBINE BLADES
BY THE FINITE ELEMENT METHOD
PART II

by

John H. Preisel, Jr.

March 1978

Thesis Advisor: Gilles Cantin

Prepared for: Naval Sea Systems Command
Washington, D.C.
Code 0331

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPS69-78-010	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Stress Analysis of Ceramic Gas Turbine Blades by the Finite Element Method Part II	5. TYPE OF REPORT & PERIOD COVERED: Engineering Thesis; March 1978
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): John H. Preisel, Jr.	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: N0002478WR8G192
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School Monterey, California 93940	12. REPORT DATE: March 1978
	13. NUMBER OF PAGES: 132
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Naval Postgraduate School Monterey, California 93940	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Gas Turbine, Stress Analysis, Finite Element; Pre- and Post-Processor

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The complex geometry of many parts of a ceramic gas turbine engine and the need to know the stresses in these parts to a high degree of accuracy indicate that the finite element method should be employed. One analysis code used a finite element program based on an isoparametric 12-node brick. This allowed quadratic variation in one coordinate direction, but only linear variation in the other two coordinate directions. This intro-

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |
(Page 1)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

duced an artificial stiffness into the structure. This study was made to determine the effects of using a totally quadratic isoparametric brick, vice one which is quadratic in only one direction.

A distributed load preprocessor was also developed and tested.

FORTRAN IV was used throughout.

Approved for public release; distribution unlimited.

Stress Analysis of Ceramic Gas Turbine Blades by the Finite Element Method Part II

by

John H. Preisel, Jr.
Lieutenant, United States Navy
B.S.Mar.E., United States Naval Academy, 1972

Submitted in partial fulfillment of the requirements for the degrees of

MECHANICAL ENGINEER

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1978

ABSTRACT

The complex geometry of many parts of a ceramic gas turbine engine and the need to know the stresses in these parts to a high degree of accuracy indicate that the finite element method should be employed. One analysis code used a finite element program based on an isoparametric 12-node brick. This allowed quadratic variation in one coordinate direction, but only linear variation in the other two coordinate directions. This introduced an artificial stiffness into the structure. This study was made to determine the effects of using a totally quadratic isoparametric brick, vice one which is quadratic in only one direction.

A distributed load preprocessor was also developed and tested.

FORTRAN IV was used throughout.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to thank all of the people who assisted me in my pursuit of knowledge at NPS.

In particular, I would like to thank Professor Gilles Cantin for introducing me to the finite element method, and his guidance throughout our association.

The cooperation of the AiResearch Division of the Garrett Corporation must also be acknowledged. Mr. D. J. Tree, and Mr. H. S. Kington were especially helpful.

A special thanks goes to my wife Marybeth for her constant help, encouragement and companionship. Without her, this and many other projects would never have been undertaken.

I. INTRODUCTION

A. WHY CERAMICS?

There are three major advantages in using ceramics in the hot section of a gas turbine: improved engine performance, higher durability, and less dependence on expensive imported materials.

Simple Brayton cycle analysis shows that a higher turbine inlet temperature (TIT) results in a higher thermal efficiency, and lower specific fuel consumption. Blade material considerations usually limit TIT. Intricate blade cooling schemes, such as those used in the LM2500 and the FT9, use impingement and convective heat transfer and external film cooling, to lower the temperature of the turbine blade. This degrades engine performance and introduces manufacturing complexities. Additionally, the cooling holes are susceptible to plugging, which may lead to eventual failure.

Ceramics, however, can withstand higher turbine inlet temperatures without blade cooling. Ceramic components also have significantly higher resistance to hot corrosion than metal "superalloy" engines. The corrosion properties of ceramics make them better candidates to withstand the higher fuel impurities expected in the future.

At present, metal engines require expensive, imported metals such as chromium, cobalt, columbium, and nickel.

These metals are in limited supply and come from areas of potential political instability. Ceramics rely on inexpensive raw materials which are found in the United States.

B. DESIGN CONSIDERATIONS

Ceramics are brittle materials. This does not mean that they are necessarily fragile. A brittle material will fracture with little or no plastic deformation. A fragile material will fracture at low stress with little or no plastic deformation. It has been suggested that the word "inductile" be used instead of brittle [1]. The inductile nature of ceramics is the crux of the design problem with this material. Local stress concentrators, such as the tip of a crack, will not yield in plastic deformation. If the stress concentration is not relieved, material fracture may result.

Another major problem is the variability of data on the material properties of ceramics.

These two problems have a major impact on the design. Design procedures that were valid for most engineering metals may fail when applied to ceramics.

C. PROJECT BACKGROUND

Some of the calculations performed later in this work are concerned with the design of a ceramic engine for a Defense Advanced Research Projects Agency (DARPA) project. The prime contractor, the AiResearch Division of the Garrett Corporation, used their T76 model metal engine as

the basis for the ceramic design. They indicated that by replacing certain metal hot section parts, they could improve engine output power by 40%, and reduce specific fuel consumption by 10%, as shown in Fig. 1. An engine cross-section (Fig. 2) shows the ceramic hot section.

As mentioned earlier, the brittle nature of ceramics drives the design. In the case of the Garrett engine, a compliant layer was added in the contact region of the dovetail as an attempt to reduce the contact stresses. Blade tip clearance was increased to reduce the possibility of impact of the blade on the shroud. In addition, a probabilistic approach, based on the Weibull distribution, was used as part of a design methodology that was aimed at predicting probability of success. These are but a few of the examples of how ceramics affected the design.

D. REASON FOR THESIS

The complex geometry of many of the ceramic engine's parts, and the need to know the stresses to a high degree of accuracy, indicated that the finite element method should be employed. One analysis code used a finite element program based on an isoparametric 12-node brick. This allowed quadratic variation in one coordinate direction, but only linear variation in the other two coordinate directions. This introduced an artificial stiffness into the structure. This study was made to determine the effects of using a totally quadratic isoparametric brick, vice one which is quadratic in only one direction.

II. THEORY

Recall from strength of materials:

$$M = EI/\rho$$

where M = moment, E = Young's modulus, I = moment of inertia, and ρ = radius of curvature.

Consider a beam made of an isotropic material, with a constant cross-section. If the beam is in pure bending, the moment is constant across the span. If Young's modulus and moment of inertia are also constant, the radius of curvature is a constant. A curve which has a constant radius of curvature is an arc of a circle.

Figure 3(a) shows a 4-node plane element. There can only be linear variation on each of the sides. If the element is deformed in pure bending, each of the sides remains straight. The top and bottom surfaces do not reflect any of the curvature expected. This is a very stiff approximation.

A 6-node plane element, shown in Figs. 3(b) and 3(c), yields a better approximation to the bending phenomenon, depending on the orientation of the one quadratic side. Orientation 1 does not reflect any of the bending curvature. Orientation 2, however, does. The finite element method would represent the deflected curve, which is defined by 3 nodes, by passing a parabola through the node points. This

can give a very close approximation to the circular arc predicted by strength of materials. It should be noted, however, that orientation 2 is constrained by the same "planes remain plane" restriction applied to slender beam theory. Thus, depending on the geometry and loading, a 6-node plane element with orientation 2 may or may not accurately reflect bending for a deep beam.

A totally quadratic element, shown in Fig. 3(d), is not constrained by orientation, geometry or loading. An 8-node plane element can give a better representation of bending.

For a given material, a stiffer element, such as the 4- or 6-node plane element, will deform less than it should. Since strain is based on displacement, and stress is based on strain, the stiffer structure will predict a lower stress than is actually present. This is a potentially dangerous design situation.

The development presented here is based on a two-dimensional plane element. Similar reasoning can be applied to three-dimensional bricks. A 12-node brick, for example, has only one quadratic side. Depending on the loading, geometry, and orientation of the quadratic side, a 12-node brick could approach the correct solution. It will always, however, be stiffer than the actual structure. If it is oriented incorrectly with respect to the applied moment, it could result in seriously incorrect answers for a coarse mesh system. In addition, since two of the coordinate

directions are stiffer than the third quadratic coordinate direction, an artificial anisotropy is introduced into the problem.

A fully quadratic brick does not have the problems mentioned above. Although the discretization process introduces some artificial stiffness, it is very slight. A fully cubic brick would even be a better representation of the actual structure.

III. BENDING SPECIMEN PROBLEM

To evaluate the accuracy of a finite element computer program, it is often useful to compare its results to experimental results for a problem that can be verified experimentally. A 4-point flexure test has been used very often for this purpose [3]. This chapter explains how the flexure test can be modeled using 12-node and 20-node bricks. The results are also discussed.

A. FOUR POINT FLEXURE TESTS

Figure 4 shows a schematic of a 4-point flexure rig. It is used to provide test results from a specimen in pure bending. Figure (5a) shows a free body diagram for the specimen. Figures (5b) and (5c) show the shear and moment diagrams. The moment diagram shows that the portion of the span between the applied forces on the top surface has no shear, and maximum constant moment, i.e., pure bending.

B. MODELING TECHNIQUES

A standard 0.125 inch x 0.250 inch x 3.0 inch bend specimen was used. The specimen was modeled using a variably-noded three-dimension brick from the ADINA [4] and SAP IV [5] finite element programs. Maximum use was made of the symmetry in the problem. Only 25% of the specimen needed to be modeled, as shown in Figs. 6a,b,c. Symmetry in the spanwise and long tranverse directions was used.

A cross-sectional view of the deformed bar, Fig. 7, shows the symmetry in the long transverse direction. The mid-span cross-section was given a guided end condition (no displacement in the x-direction). The midplane in the long transverse direction was also given a guided end condition (no displacement in the y-direction).

Since the model was very regular in shape, automatic nodal point coordinate and connectivity generation were extensively used. Both were generated in the spanwise direction to take maximum advantage of the automatic generation features of the codes employed.

C. FLEXURE CASE STUDY

A 12-node brick has one quadratic direction. The other two directions are linear. When used to model the bending specimen, three distinct orientations of the brick can occur. Figure 8 shows each of these orientations with respect to the bending moment. An analysis was performed using the SAP IV program for each of the three orientations, called Case 1, Case 2, and Case 3. The applied forces were modeled as distributed loads.

1. Analysis of Results

The three-dimensional finite elements used to solve the problem would give a solution converging to an exact three-dimensional elasticity theory solution. However, such a solution in closed form does not exist. Therefore, the results were compared to the usual strength of materials beam theory. Such a theory is not perfect and circumspection

had to be used in such a comparison. For instance, beam theory does not predict any anticlastic curvature [18], but the finite element solution shows some. However, due to the boundary conditions and symmetry, beam theory should apply to the x-z plane of symmetry with a very good accuracy. The displacements of the neutral axis of this plane from x = 0 to x = L were plotted versus the span length. Displacements were normalized by dividing them by the mid-span (x = 0) displacement predicted by beam theory (δ_A). This facilitates comparisons in terms of the percentage of the maximum deflection of beam theory.

Two loadings and materials were used. The first specimen was made of steel, and had a 300 lb load. The second specimen was hot pressed silicon nitride (HPSN), and had a 180 lb load. The mechanical properties of HPSN are given in Ref. [6].

The deflection curves were calculated using [7]:

$$\delta = \delta_A + \frac{M_A x^2}{2EI} - \frac{w\langle x-a\rangle^3}{6EI}$$

where $$\delta_A = -\frac{w(L-a)}{6EI}(2L^2 + 2aL - a^2)$$

$M_A = w(L-a)$ = reaction end moment

w = applied force

E = Young's modulus

I = moment of inertia

Figure 9 shows that L is half the distance between supports on the lower side of the specimen, and w is the total load applied, i.e., 300 lb or 180 lb. The symbol < > denotes a

discontinuous function, i.e., for $x<a$ the quantity $\langle x-a \rangle = 0$, for $x \geq 0$ $\langle x-a \rangle = x-a$.

The results of these analyses are shown in Figs. 10 and 11. The Case 1, 2 and 3 curves correspond to three orientations of the 12-node brick. Case 1, the orientation with the quadratic side along the axis of bending, is the closest of the three to the values predicted by beam theory. The maximum deflection of the ceramic specimen is approximately 96% of the deflection predicted by beam theory. This model is slightly stiffer than beam theory. Cases 2 and 3, which have the quadratic side perpendicular to the axis of bending, are significantly stiffer models than Case 1. The maximum deflection predicted by these cases is only 70% of the deflection predicted by beam theory.

In the displacement-based finite element method, stress and displacement are related by [8]:

$$\{e\} = [B]\{u\}$$

and

$$\{\tau\} = [C]\{e\} = [C][B]\{u\}$$

where $\{\tau\}$ = a vector composed of the six independent components of the stress tensor.

$\{e\}$ = a vector composed of the six independent components of the small strain tensor.

$[B]$ = strain-displacement transformation matrix.

$\{u\}$ = displacement matrix.

$[C]$ = matrix of material properties.

This formulation clearly shows that a stiffer structure (i.e., less displacement and displacement gradients) will produce less stress.

Case 4, shows results which are slightly more flexible than beam theory. Since beam theory is inherently stiff due to the approximations made during its development (i.e., negligible shear stress), the Case 4 model is closer to predicting the actual stress distribution than either Cases 1, 2, 3 or beam theory. It should also be remembered that the actual structure modeled will bend in all three directions. Clearly, the only way to fully represent such a deformation is to use a fully quadratic element. Any element that is less than fully quadratic will be too stiff.

D. TEMPERATURE GRADIENT CASE STUDIES

The temperature gradient case studies used the same models as the flexure case study. Cases 1, 2, 3 refer to the same orientation of the 12-node brick as in the flexure study. Case 4 is the 20-node brick model.

Hot pressed silicon nitride was used as the specimen material. The coefficient of thermal expansion is given in Ref. [6]. A 100°F temperature gradient was imposed across the short transverse (0.125 inch). The stress-free reference temperature was 1500°F. A variably-noded (12-21) three-dimensional brick from the ADINA finite element program was used to model the problem. Nodal temperatures were input via a temperature tape. This procedure is described in Appendix A.

As in the flexure case study, the normalized deflections of the y-z plane of symmetry were plotted versus distance along the span. Beam theory was used as the basis for comparison. The deflection predicted by beam theory is [7]:

$$\delta = \delta_A + \frac{\alpha}{2t} (T_2 - T_1) x^2$$

where: $\delta_A = -\alpha L^2 (T_2 - T_1)/2t$

α = coefficient of thermal expansion (inch/inch/$^{\circ}$F)

t = depth of beam

$(T_2 - T_1)$ = temperature difference through depth of beam

L = half the distance between the lower reactions

Two different case studies evolved from the temperature problem. Reference [9] states that the Bernoulli-Euler assumptions can be used in practical analyses of beams under thermal loadings. The assumptions state, in part, that the effects of lateral contraction can be ignored, i.e., Poisson's ratio (ν) may be taken as equal to zero. In the case studies that follow, Poisson's ratio was used in the first study, and set equal to zero in the second.

1. Results of the Case Study using Poisson's ration (ν)

The normalized deflection curves for this case study are shown in Fig. 12. The displacements from x=0 to x=L are plotted. Cases 2 and 3 are stiffer than beam theory. The curves show that the maximum deflection calculated for these two cases is 75% of the maximum deflection predicted by beam theory.

The results of Case 4, which used the fully quadratic 20-node brick, show excellent agreement with simple theory.

Figure 12 shows that the results of Case 1 (the 12-node brick with quadratic side along the span) predict a deflection greater than theory. It is puzzling that the results show that the element is even more flexible than the 20-node brick. The temperature algorithm used by ADINA was checked extensively. No errors were detected. No satisfactory explanation for the 12-node brick behavior has been found. It is suspected that the inherent anisotropy of the 12-node brick may be the cause of the problem. However, extensive further examination would be necessary to resolve the dilemma.

Possibly more interesting than the displacement results are the stresses calculated. Recall that if a linear temperature gradient were imposed on a bar that had no external constraints, the bar would deform into an arc of a circle. Also there would be no stress in the bar according to a fundamental theorem of elasticity. In this case study, the bottom nodes at $x = 0.75$ inch were fixed in the z-direction. The part of the specimen to the left of this constraint (i.e., toward the free end) should, however, show no stress until one reaches the immediate vicinity of the constraint. Each of the cases that use the 12-node brick predict stresses, both normal and shear, on the order of 10^3 to 10^4 PSI in the free end. The 20-node brick, however, predicts stresses of approximately 5 PSI or less in this region. ($E=45\times10^6$ psi, $\nu=0.2$, $\alpha=1.7\times10^{-6}$ in/in/^{o}F).

The above discussion concerning the 12-node brick models indicates the possible uncertainty that can result from using this element in thermal calculations.

2. Results of Case Study with $\nu = 0$

This case study was undertaken in an attempt to explain the results of the Case 1 orientation of the 12-node brick discussed above. Everything was the same as in the previous study, except that the effects of Poisson's ratio were neglected. The results are shown in Fig. 13. The Case 2 and 3 models have become significantly stiffer. The maximum deflection was approximately 67% of the value predicted by theory, as compared to approximately 75% when Poisson's ratio was used. The results for Case 4 have remained the same (to three significant digits). The displacements for Case 1 have decreased to such a degree that they were almost identical to Case 4. The spurious free end stresses cited previously remained.

In summary, by ignoring the effects of Poisson's ratio, three of the models have become significantly stiffer, while the fully quadratic model has remained substantially the same and in close agreement with the theory.

It is interesting to note that if the specimen were truly unconstrained and had a linear temperature gradient, the results of analyses that take into account Poisson's ratio should agree exactly with the results of analyses that do not [9].

IV. AIRFOIL MODELING

This chapter discusses the techniques developed to construct a finite element mesh for the airfoil section of a ceramic gas turbine blade. This particular engine airfoil lent itself to detailed analysis because of the relative ease with which the mesh could be generated. This is due to the linear fairing between the tip and root sections of the airfoil. Although linear fairing was not optimal from an aerodynamic point of view, it was desirable from a manufacturing point of view. Its relative ease of fabrication caused it to be used in practice.

A. DEVELOPMENT OF AIRFOIL PROFILES

The Garrett Corporation provided Mylar cross-sectional profiles of the first stage rotor blade of their experimental ceramic turbine [10]. These profiles were for various radial distances as measured from the axis of rotation along the stacking axis of the blade. In addition to an outline of the cross-section and its relation to the stacking axis, a table of x-y coordinates was given for the profile. These coordinates described 62 points on the profile.

The 62 profile points were assembled into x and y matrices. These matrices were used to develop cross-sectional plots on the time sharing terminal and later on the Calcomp plotter. Figures 14 and 15 show representative cross-

sections of the tip and hub sections. The profile points are marked by an 'x' and are connected by straight line segments.

Since there was linear fairing between the tip and hub, it could be easily verified that the coordinates listed on the Mylar drawings were given at the same relative positions along the foil profile. In other words, the ith point on the R=3.1 inch profile corresponded to the ith point on the R=4.0 inch profile, where R is the radial distance along the stacking axis from the axis of rotation.

The root profile was then divided into elements. Care was taken to develop elements with convex corners. Also, a relatively fine mesh was desirable in the leading and trailing edge areas of the airfoil to enable accurate modeling of the anticipated stress concentrations there. The resulting leading and trailing edge elements proved quite troublesome; this will be discussed more fully below.

The mesh developed in the airfoil root section dictated the mesh in the tip section. This was due to the unique correspondence of points in the tip and hub sections. The tip section was then connected to the hub section by linear nodal coordinate generation. Connectivities were generated from the leading to the trailing edge. This scheme took maximum advantage of the automatic generation features of SAP IV and ADINA.

B. THE NEGATIVE JACOBIAN DETERMINANT

As mentioned earlier, the leading and trailing edge elements were ill-behaved. The leading and trailing edges of

this blade are arcs of a circle. When an element with quadratic sides is used, these elements are prone to re-entrant angles, and resultant negative or zero determinants of the Jacobian matrix. In other words, the inverse of the Jacobian matrix does not exist because there is not a unique correspondence between the natural and local coordinates of the element [8]. This will stop the program because the inverse of the Jacobian matrix is used in the strain-displacement matrix [B]. Recall:

$$\{\varepsilon\} = [B]\{u\}$$

$$[B] = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} [H]$$

$$\begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{Bmatrix}$$

Also: $[K] = \int_{VOL} [B]^T [C] [B] \, d(VOL)$

$[J]^{-1}$ = inverse of the Jacobian matrix

$[K]$ = stiffness matrix

$\{u\} = [H]\{u_i\}$ - displacement field

$[H]$ = matrix of shape functions

$[C]$ = matrix of elastic properties

$\{\varepsilon\}$ = strain matrix

r,s,t = natural coordinates

As the above equations show, the inverse of the Jacobian is an essential part of the calculations for the strain matrix, and the stiffness matrix. Experience has also shown that when the determinant of the Jacobian is small (10^{-7} or smaller) the displacements at that node may lose almost all their numerical significance.

C. MESH VERIFICATION THROUGH PSAP1

Kibler [11] implemented a finite element preprocessor for mesh verification for the SAP IV and ADINA programs. This enables the user to check visually the coordinates and geometry of a mesh by generating graphical displays of the completed mesh. Oblique orthographic projections are produced using the NPS Calcomp model 765 plotter. Other displays are possible.

The PSAP1 program is especially useful because of the various mesh presentation options in the program. Among them are:

1. Node numbering
2. Element numbering

3. Exploded plots
4. Full rotation about each axis
5. Sectioned plots based on geometry, node number, or element number
6. Displacement postprocessing using either the deformed structure, or displacement vectors at the nodes. (See Chapter VII)

Other options, as well as a User's Manual, are fully discussed in Ref. [11].

Figures 16 through 19 show the various mesh presentations, for the linear airfoil sections. PSAP1 was used extensively in the production of the airfoil mesh. The clear graphical presentation assures the engineer that he is using a mesh free of geometric or connectivity errors.

The elements with negative Jacobians were separately selected from the mesh and greatly magnified. The re-entrant angles were clearly visible on these plots as shown in Fig. 20. Large scale plots of the hub and tip cross-sections for these elements were made by hand. The mid-size nodes were moved inward toward the geometrical center of these elements.

Two airfoil meshes were developed. One was composed of 20-node bricks, and the other of 12-node bricks. The 12-node bricks were oriented such that the quadratic side could be used to model the airfoil camber. This orientation was the same as the one used by Garrett. Since the 12-node brick has only one quadratic side, part of the leading and trailing edge curves were eliminated. This is shown in Fig. 21. Aside

from this difference, the geometry represented by the two meshes was identical. The 12-node airfoil is shown in Fig. 22.

The characteristics of each of these meshes are shown in Table I.

V. AIRFOIL CASE STUDY

After thoroughly investigating the ceramic bar problem, the next step was to analyze linear airfoil section of the ceramic gas turbine blade using the ADINA code 12-node and 20-node bricks. The bar problem showed that significant differences occurred as a result of the brick type and orientation used. The purpose of the airfoil study was to determine the difference, if any, between the 12-node brick airfoil and a 20-node brick airfoil. Each airfoil was loaded using the actual design point conditions that could be expected in the ceramic gas turbine. Two general types of loading were investigated: centrifugal and thermal.

A. CENTRIFUGAL LOAD

A centrifugal load preprocessor was developed at NPS, and is discussed fully in Ref. [12]. The preprocessor can be used to generate consistent nodal loads for a given angular speed. The rotor design speed of 41,730 RPM was used.

Figures 23, 24 show the deformed foil superposed on the undeformed foil. (Chapter VII discusses using PSAP1 to plot the deformed structure.) The root plane coincides exactly in all cases because the nodes there were fully constrained. Two general deformations can be seen: first, there is a general elongation of the blade in the radial direction, and

second, there is an untwisting of the blade. Figures 25-26 show the deformed 12-node and 20-node foils superposed on each other. It can be seen that the 12-node model is stiffer in representing the untwisting deformation. Tables II and III show comparisons of stresses and maximum principal stresses, as calculated by the 12-node and 20-node bricks, at selected nodes in the root section. At each of the tabulated nodes, the stresses predicted by the 20-node brick are higher than those predicted by the 12-node brick. The stress contours for the root section are shown in Figs. 29 and 30.

B. THERMAL LOAD

Reference [13] shows the steady state isotherms. For this study, the isotherms were modeled as constant for various radial distances along the foil. The cited reference shows that this is a good assumption in the root section, where the isotherms are flat and closely spaced. In the upper portion of the foil, however, the isotherms have a definite curvature in the radial direction. Figure 31 shows the nodal temperatures used in this study. Preparation of the nodal temperature tape is discussed fully in Appendix A.

Figure 32 shows the deformed foil superimposed on the original geometry. A general enlargement of the foil in all directions is seen. Table IV shows a comparison of thermal stresses at selected points. (It should be noted that these are not evaluated at nodal points, but rather at the Gauss points. ADINA does not give nodal stresses for thermal loadings.)

C. CONCLUSIONS FROM AIRFOIL STUDY

This comparative study shows how the choice of a finite element can affect the results. The 12-node brick, even though oriented in the most advantageous way, is stiffer than the fully quadratic three-dimensional element. The stiffer element produces lower stresses in the root section. It is these root section stresses which transmit the tensile forces and moments to the blade platform and eventually to the critical dovetail area.

VI. DISTRIBUTED LOAD PREPROCESSOR

The version of ADINA installed at NPS does not contain a distributed load feature. Finite element texts, such as Ref. [14], often state a consistent allocation of a uniform surface load for a rectangular element. The consistent load for a curved surface, such as the airfoil studied in this report, is more complex. Since a distributed load is used to model the aerodynamic pressure on an airfoil, a program to calculate consistent pressure loads for curved surfaces was necessary.

A. DEVELOPMENT OF THEORY

A consistent load is defined as a set of nodal forces which produce the same amount of work as the original force. Stated mathematically:

$$\langle v_i \rangle \{u_i\} \equiv \int_{AREA} \langle f^S \rangle \{u\} \, d(AREA)$$

where: $\langle v_i \rangle$ = vector of nodal consistent loads

$\{u_i\}$ = nodal displacements

$\langle f^S \rangle$ = vector of surface forces

$\{u\} = [N_i]\{u_i\}$ = displacement field

$[N_i]$ = shape functions

Consider the general surface shown in Fig. 33.

$$\text{Work} = \int_A \underline{n} \, pdA \cdot \underline{u} = \langle v_i \rangle \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \qquad (1)$$

$\underline{n}$ = unit normal to the surface (positive outward).

p = pressure on dA.

$\underline{u}$ = displacement vector = $u\underline{i} + v\underline{j} + w\underline{k}$

For ease of computation, it is desirable to perform the integration in the natural coordinate system. The vector $\underline{n}dA$ is transformed to the natural coordinate system by:

$$\underline{n}dA = \left(\frac{\partial \underline{R}}{\partial \alpha} \times \frac{\partial \underline{R}}{\partial \beta}\right) d\alpha \, d\beta$$

where: $\underline{R} = x\underline{i} + y\underline{j} + z\underline{k}$ = position vector of point on surface as shown in Fig. 33.

$d\alpha \, d\beta$ = differential area in natural coordinate system.

α, β are dummy variables standing for the appropriate natural coordinates which define the pressure face (See Table V).

$\frac{\partial \underline{R}}{\partial \alpha}$ This is a vector tangent to a line α where β is constant. The components of this vector are elements of the Jacobian matrix.

$\frac{\partial \underline{R}}{\partial \beta}$ is similarly defined.

The pressure and displacement fields can be discretized with the same shape functions:

$$p = \langle N_i \rangle \{p_i\} \qquad (2)$$

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \qquad (3)$$

where: N_i = shape functions

$\{p_i\}$ = matrix of nodal pressures

$\begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}$ = matrix of nodal displacements

Let:

$$\left(\frac{\partial \underline{R}}{\partial \alpha} \times \frac{\partial \underline{R}}{\partial \beta}\right) d\alpha\, d\beta = (\lambda_1 \underline{i} + \lambda_2 \underline{j} + \lambda_3 \underline{k})\, d\alpha\, d\beta \qquad (4)$$

then:

$$\underline{n} dA \cdot \underline{u} = \langle \lambda_1\ \lambda_2\ \lambda_3 \rangle \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} d\alpha\, d\beta$$

$$\lambda_1 = x_{2,\alpha}\, x_{3,\beta} - x_{3,\alpha}\, x_{2,\beta}$$

$$\lambda_2 = x_{3,\alpha}\, x_{1,\beta} - x_{1,\alpha}\, x_{3,\beta}$$

$$\lambda_3 = x_{1,\alpha}\, x_{2,\beta} - x_{2,\alpha}\, x_{1,\beta}$$

Using Eqs (2 through 5), Eq (1) can now be written

$$\langle v_i \rangle \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} = \int_{-1}^{1} \int_{-1}^{1} \langle \lambda_1 \lambda_2 \lambda_3 \rangle \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \langle N_i \rangle \{p_i\} d\alpha\, d\beta$$

$$\langle v_i \rangle = \int_{-1}^{1} \int_{-1}^{1} \langle \lambda_1 \lambda_2 \lambda_3 \rangle \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \langle N_i \rangle \{p_i\} \, d\alpha \, d\beta \qquad (6)$$

B. PROGRAM STRUCTURE

The consistent pressure load program is structured as follows:

Subroutine TRANS

- reads element number, coordinates, and connectivity
- reads the face to which pressure applied (NFACE)
- reads pressure at each node of NFACE
- reads number of Gauss points to be used in surface integration (2-6)
- calls CUBAT

Subroutine CUBAT

- establishes vector of weights and sampling points to be used in Gaussian integration
- calls SHAPE

Subroutine SHAPE

- evaluates shape functions at Gauss points
- evaluates derivative of shape function at Gauss point
- forms Jacobian matrix
- evaluates determinant of Jacobian matrix to check for zero or negative Jacobian determinant
- calls FACEP

Subroutine FACEP

- establishes matrices in integrand
- performs matrix multiplication to establish integrand
- performs surface integration

A listing of the consistent pressure load program is in Appendix C.

C. PROGRAM VERIFICATION

Several test cases were run to insure the accuracy of the pressure load preprocessor. A uniform distributed surface load was rotated through all faces of a one inch cube, and the 2-6 Gauss point integration scheme was checked for each face. The results agreed exactly, both in magnitude and sign, with the consistent load allocation found in Ref. [14].

The program was successfully checked to insure that distributed surface force could be applied to more than one face of the element simultaneously, and that pressures could be applied simultaneously to faces that share a common edge.

The ceramic flexure specimen mesh (Case 4) was also used to verify an actual problem. Pressure loads were applied to various faces of the specimen, including adjoining faces. In each case, the program produced correct results.

As a final test, a distributed load was also placed across the span between the supports. A sketch of the

problem is shown in Fig. 34. The results were again compared with simple beam theory. The results were excellent as shown in Fig. 35.

D. USER'S MANUAL

Figure 36 shows a hexahedral element in the natural coordinate system. The node numbering convention is also shown in Fig. 36. Table V shows the face numbering convention used in the pressure load preprocessor. Each face has four corner nodes and four midside nodes. The pressure distribution acting on an element face is defined by specifying the nodal pressure intensities at the corner and mid-side nodes. For example, if Face 4 were loaded P_4, P_8, P_7, P_3, P_{20}, P_{15}, P_{19}, and P_{11} must be specified in that order. Appendix B lists the input data and results for the sample bar problem, shown in Fig. 36.

1. Control Card (10I5)

Columns	Variable	Meaning
1-5	NUMNP	Total number of node points in problem.
6-10	NUMEL	Total number of elements in problem.
11-15	NGP	Number of Gauss points to use in surface integration. ($2 \leq NGP \leq 6$)
16-20	NCUR	Number of ADINA load curve that output loads refer to. (usually,NCUR=1)

21-25	NPLOAD	Number of element faces with distributed loads.
26-30	IPRINT 1	1-program will just read and print ADINA input deck
31-35	IPRINT 2	1-program will not print element-by-element solution.
36-40	IPRINT 3	1-program will not punch output deck of consistent loads.
41-45	IPRINT 4	1-program will omit printing input data.

2. Nodal Pressures Card (2I5, 8F5.0)

1-5	NPREL (J)	element number
6-10	NFACE (J)	face to which pressure applied. (See Table V)
11-50	PRESS (I,J)	nodal pressure intensities input here: corner nodes first, then midside. (See Table V) Positive pressure is directed in same sense as outward unit normal.

E. USE OF PRESSURE LOAD PREPROCESSOR ON CP/CMS

The pressure load preprocessor is ideally suited for use on CP/CMS. A compiled version of the program (filetype=text) should be stored on a private disk or workspace. A data file, such as FT04F002, should also be defined. The input data deck (Appendix B) can then be read into this file [15].

It is suggested that an executive type routine also be used:

```
& TYPEOUT ERROR
FILEDEF 05 DSK FILE FT04F002 (PERM)
FILEDEF 06 PRT (PERM)
FILEDEF 07 PUN (PERM)
VSET RDYMSG OFF
$ PRESSURE
```

This defines your input/output files and executes the program. Output is directed to the line printer, and card punch.

The pressure program utilizes approximately 300K BYTES of core. The extra core should be requested at LOGIN by:

```
L␣nnnnPtt␣300K
```

where:

nnnn = user number

tt = terminal number

VII. USING PSAP1 AS A DISPLACEMENT POSTPROCESSOR

PSAP1 is capable of postprocessing displacements. The results can be presented in two ways:

1. The deformed structure can be plotted. In this case, scaled displacements are added to the original geometry to produce the deformed coordinates. The deformed structure is then plotted using this new geometry.

2. The displacements can be represented as scaled vectors. The scaled displacement vectors are plotted at original geometry nodes.

The purpose of this chapter is to explain the modification of the ADINA code to store the displacements, and how to use the PSAP1 postprocessing capabilities.

A. MODIFICATION OF ADINA

Reference [16] states that a version of SAP IV had been modified to punch a deck of displacement cards from the output data. When dealing with large quantities of data, however, storage of this data on direct access storage devices is preferable. Therefore, the ADINA code was modified to store the output displacements on device 58, in addition to the usual printed output.

The flag for saving the displacements is the integer number 1 in column 80 of the ADINA master control card [4]. The flag variable name is ITP 58, and is passed in common /EAST/.

The job control language (JCL) for ADINA must also be modified to define device 58:

```
//FT58F001 DD DISP=OLD,UNIT=3330,VOL=SER=DISK0,
//            DSN=Snnnn.pppp
```

where:

nnnn is the user number

pppp is the name assigned by the user to identify the data set.

Since an unformatted write statement is used to store the data on device 58, a variable blocksize must be used when allocating disk space. Reference [17] explains this procedure.

B. MODIFICATIONS OF PSAP1

There are presently two subroutines in PSAP1 that can be used to input displacement data to PSAP1.

1. Subroutine DATA 9 was added to PSAP1 by Losh [16]. This subroutine reads displacement data from punched cards. Reference [16] discusses its use, and should be consulted.

2. Subroutine DATA 5 reads displacement data directly from device 58. A JCL card must be added to the PSAP1 JCL to define device 58:

```
//GO.FT58001 DD DISP=OLD,UNIT=3330,VOL=SER=DISK0,
//              DSN=Snnnn.pppp
```

C. USER'S MANUAL FOR DISPLACEMENT POSTPROCESSING

The PSAP1 User's Manual [11] applies with the following modifications:

a) Namelist Option

Variable, Value	Description
NUDISP = 1	x direction displacements input
NVDISP = 1	y direction displacements input
NWDISP = 1	z direction displacements input
KDATA = 5	subroutine DATA 5 used to read displacement data from device 58.
NVALUS = nnn	nnn is the integer number of displacements set to be plotted. This is equal to the number of node points.

b) Namelist Pict

KDISP = n	n=1: plot of deformed structure n=3: displacements represented by vectors at nodes.
IDMAG = n	n=1: direct magnification of displacement data by DMAGS. n=2: scaling of displacement data to a maximum value of DMAGS.
DMAGS = r	r is a floating point number which indicates the magnification of the displacements. (0.1 has been found appropriate.)

Figures 37 and 38 are examples of displacement postprocessing. The structure is the bending specimen loaded in four-point flexure.

Figure 37 illustrates the deformed structure option of PSAP1. Figure 38 shows the displacements of element 15 plotted as scaled vectors at the node points.

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

1. The 12-node brick is a stiffer model of a structure than the fully quadratic 20-node brick.

2. The accuracy of the results obtained using a 12-node brick depends on geometry, loading, and the orientation of the element.

3. The flexure problem showed that two orientations of the 12-node brick are stiffer than the third. Resulting displacements and stresses differed significantly, and in all cases were lower than the 20-node brick and theory.

4. The airfoil study showed that the maximum principal stresses predicted in the root section could differ by as much as 14.9% depending on the basic element used. The 12-node brick under-predicted the 20-node results. The use of the 12-node brick could therefore produce an unsafe design.

B. RECOMMENDATIONS FOR FUTURE WORK

As is true with many projects, this thesis raised many new questions. The following areas of future work are suggested:

1. The ADINA code should be reviewed with an eye toward streamlining it. A basic code using only two- and three-dimensional elements, and the linear isotropic material model would be a useful tool. A reduced version of ADINA

could then be installed on the time-sharing system, or other stand-alone system.

2. The Field program, developed at NPS, should be examined and expanded if necessary. This program would serve as an excellent thermal preprocessor. It should be modified to produce a nodal temperature tape that could be used in ADINA, or other code capable of using temperature inputs.

3. Aerodynamic pressure loads should be examined. A program that could calculate the nodal pressure intensities for an element would provide the necessary input information for the distributed load preprocessor.

4. Centrifugal, pressure and thermal loadings should be superposed.

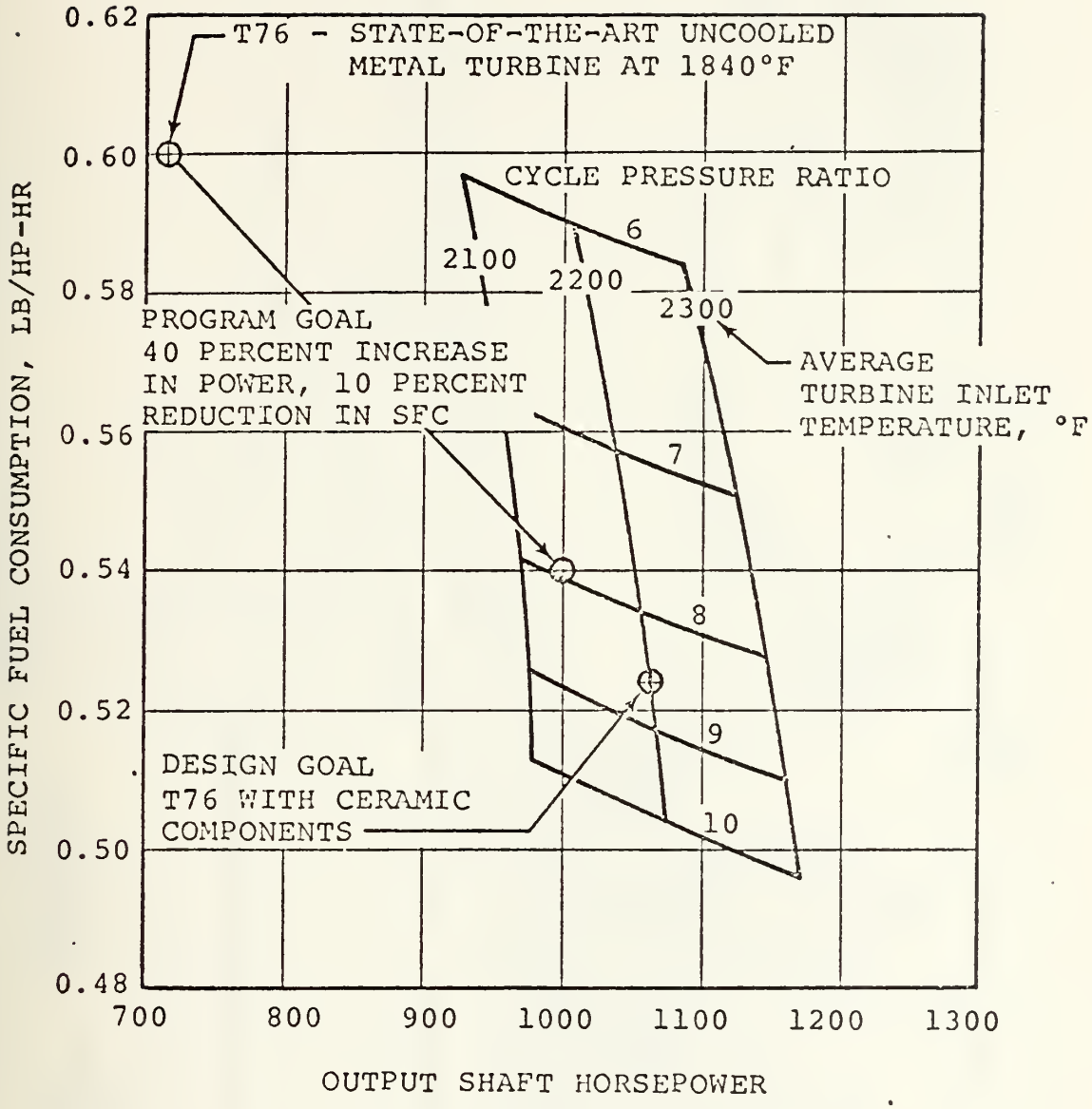


Figure 1. Engine Design Parameters

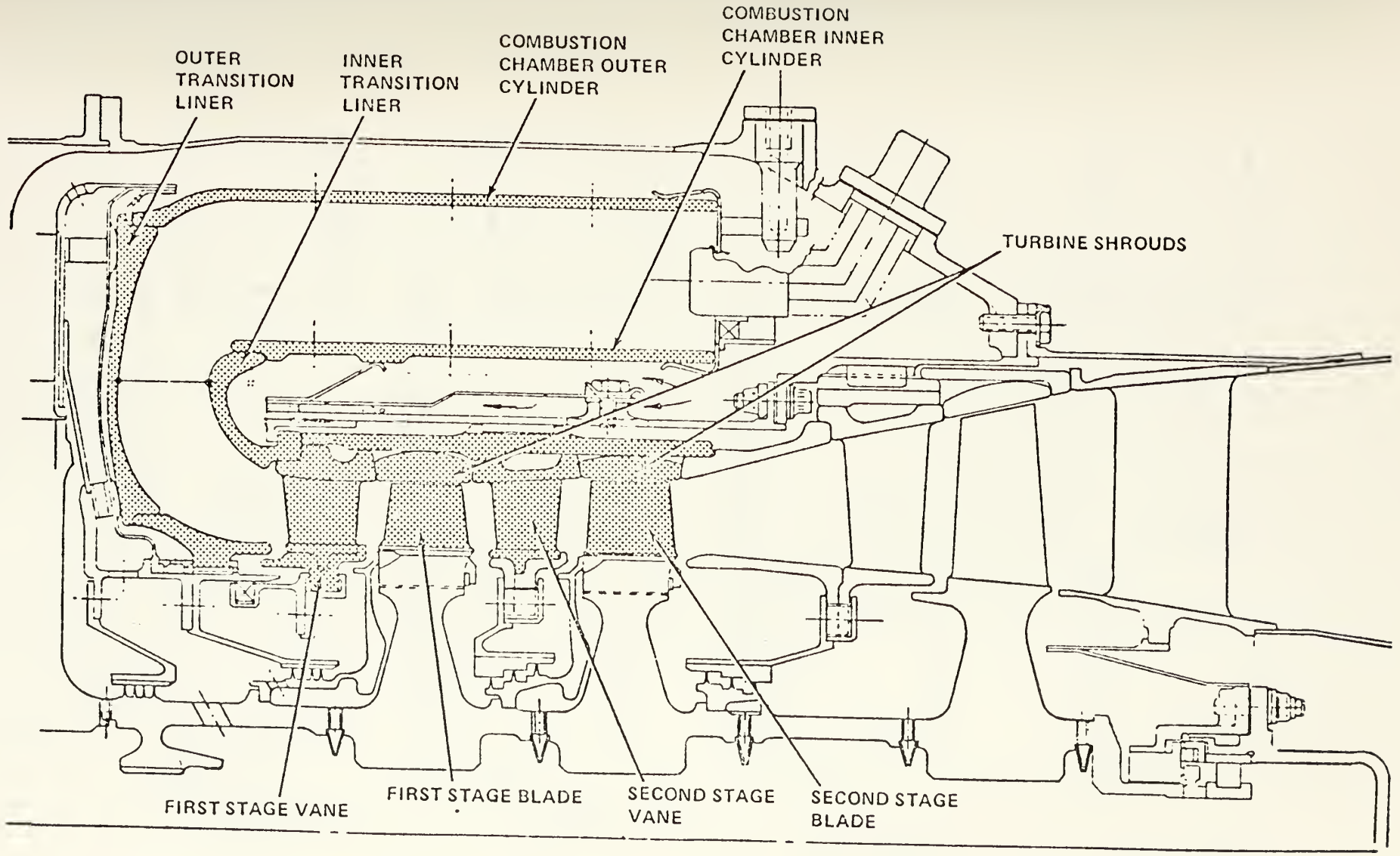
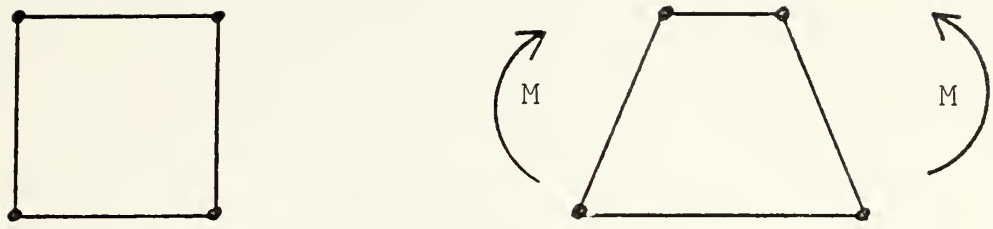
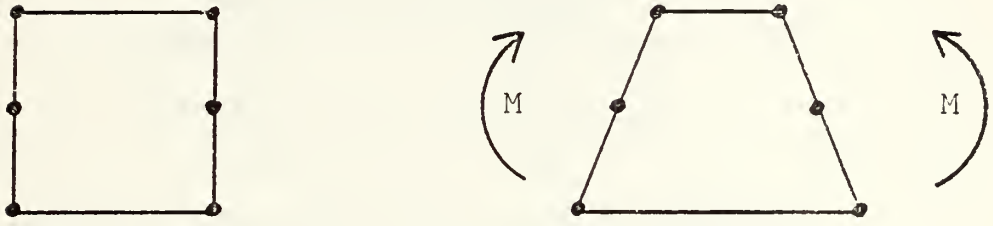


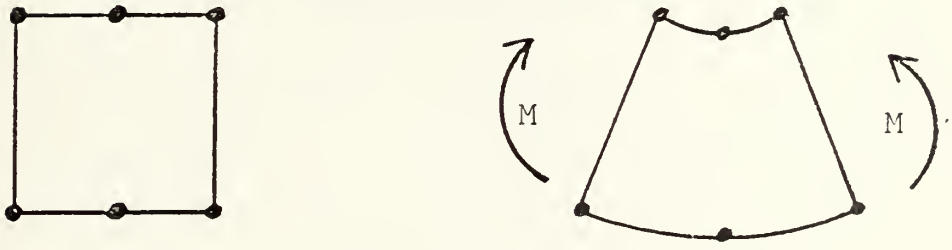
Figure 2. Ceramic Hot-Section Components



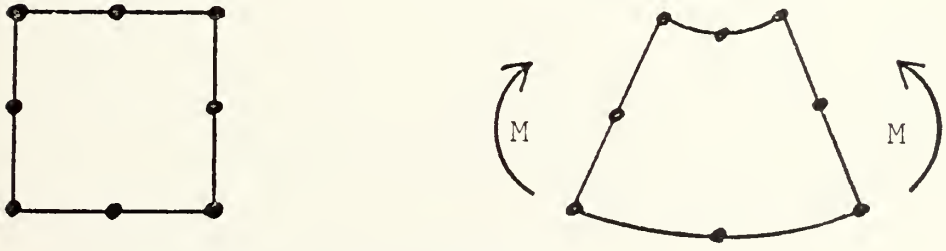
a. 4 Node Plane Element



b. 6 Node Plane Element: Orientation 1.



c. 6 Node Plane Element: Orientation 2.



d. 8 Node Plane Element

Figure 3. Two-Dimensional Plane Element

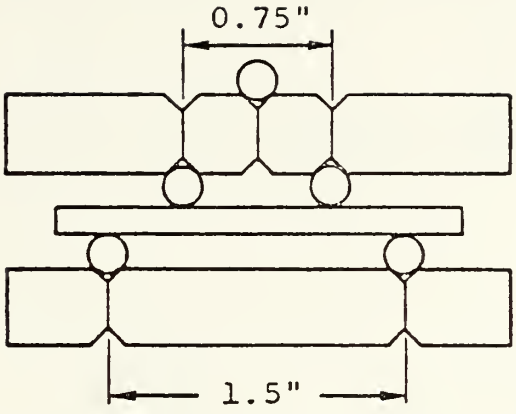


Figure 4. Schematic of 4-Point Flexure Rig

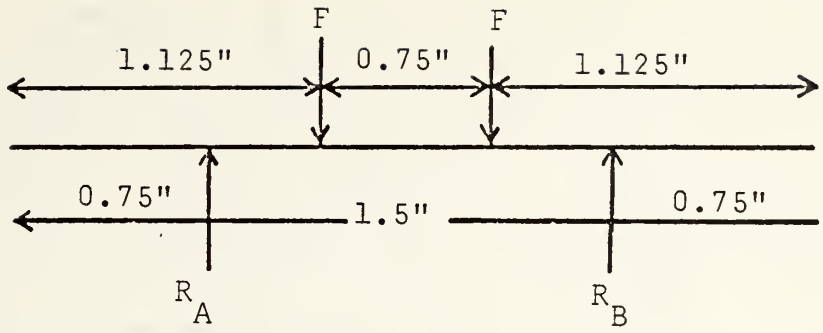


Figure 5a. Bend Specimen Free-Body Diagram

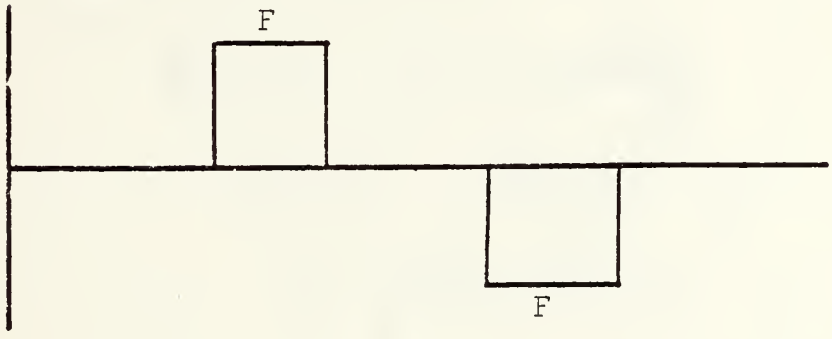


Figure 5b. Bend Specimen Shear Diagram

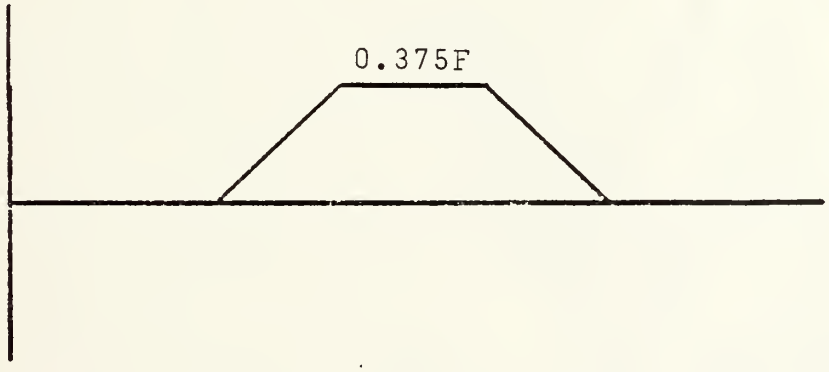


Figure 5c. Bend Specimen Moment Diagram

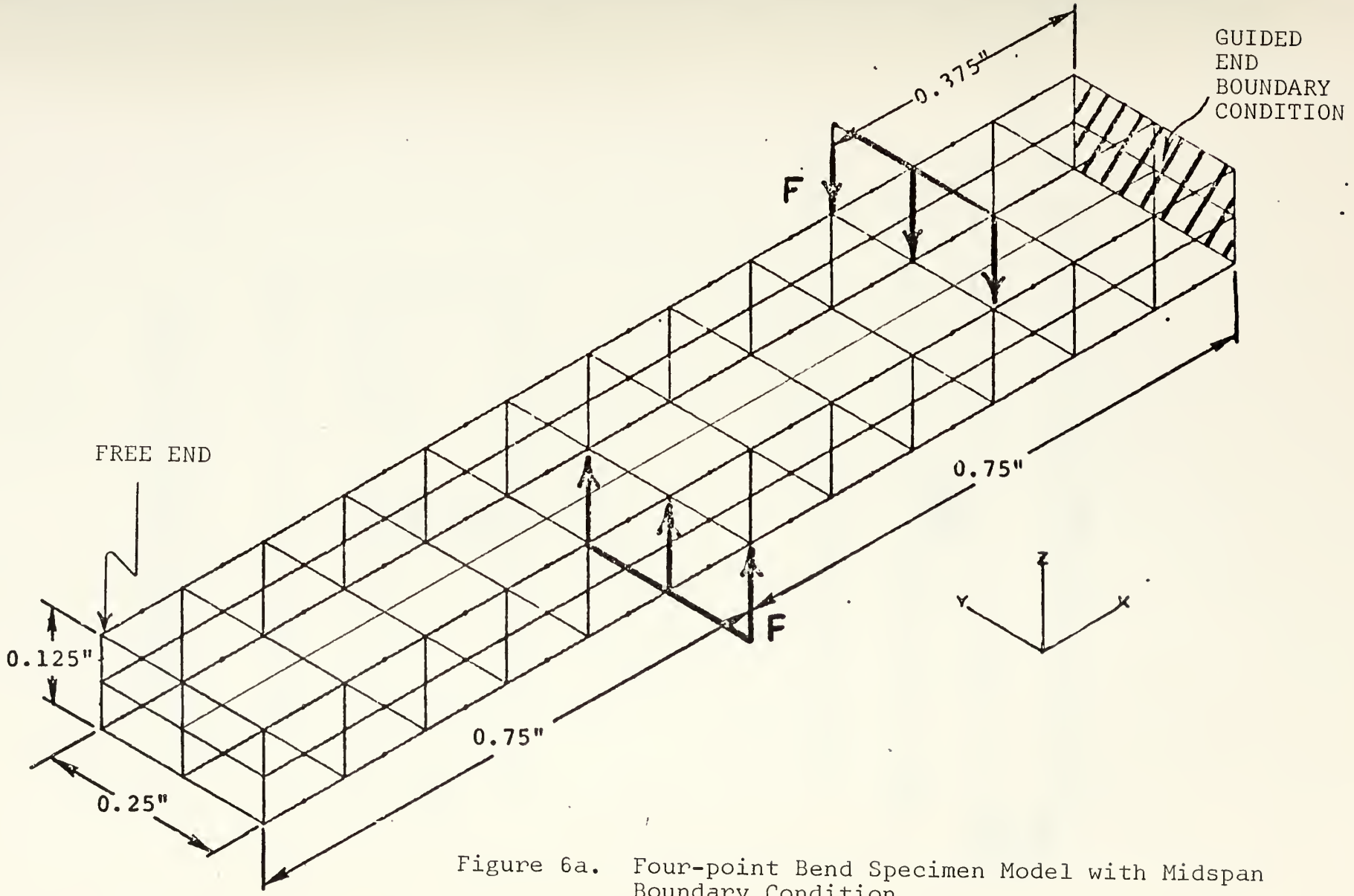


Figure 6a. Four-point Bend Specimen Model with Midspan Boundary Condition.

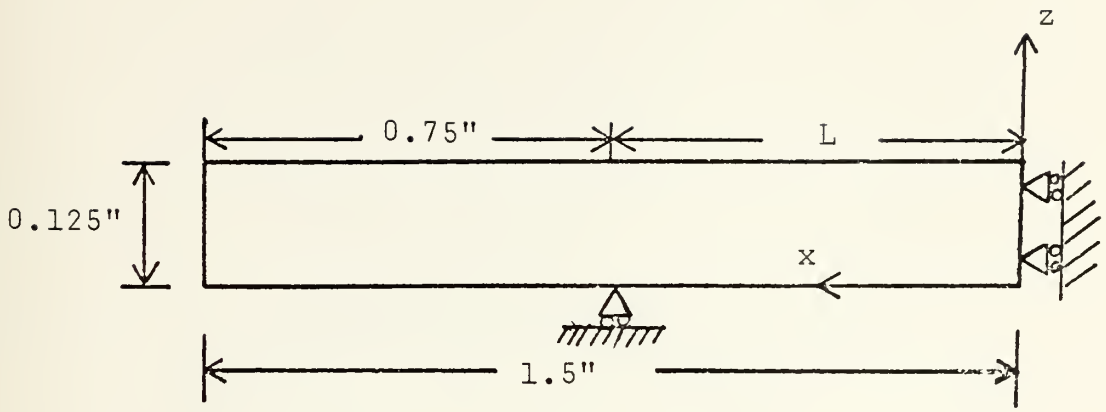


Figure 6b. x-z View of Half-Span with Guided End Condition

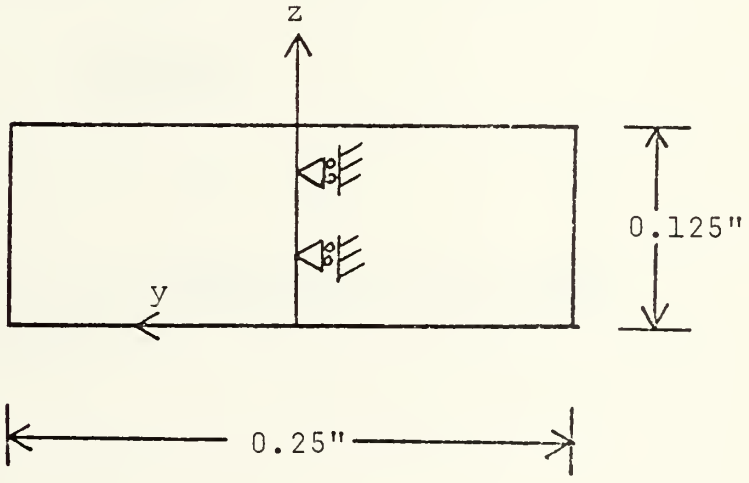


Figure 6c. y-z View of Specimen with Guided Midplane Boundary Condition

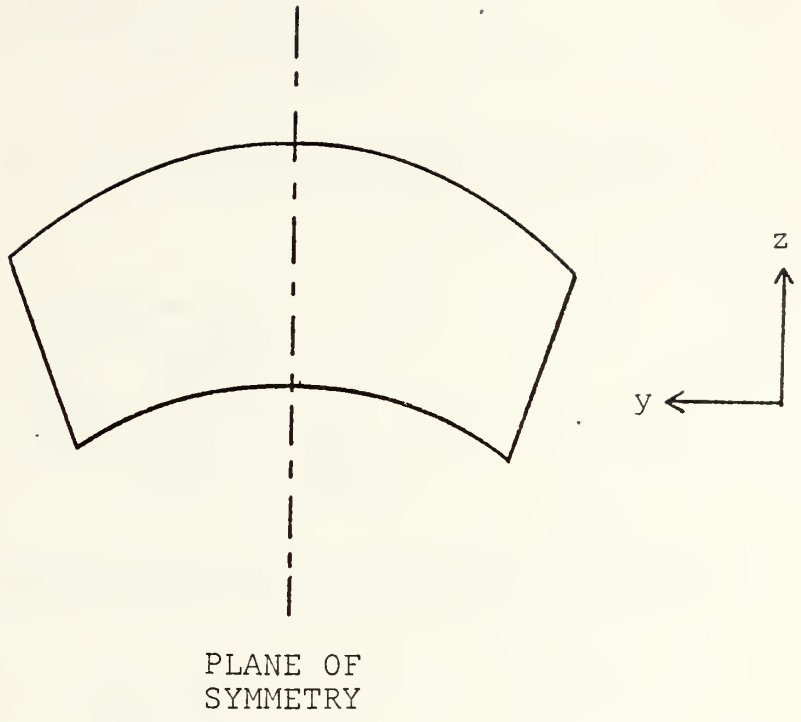


Figure 7. Cross-Sectional View of Deformed Specimen Showing x-z Plane of Symmetry

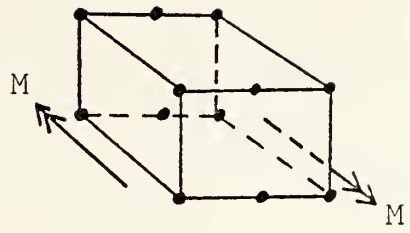


Figure 8a. 12-Node Brick Orientation for Case-1

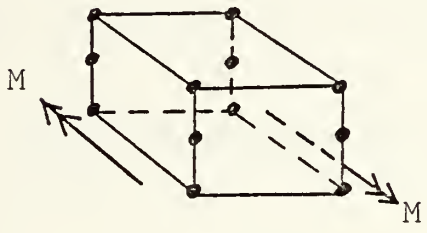


Figure 8b. 12-Node Brick Orientation for Case-2

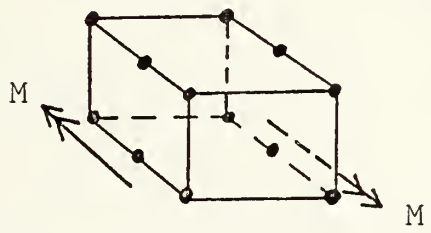
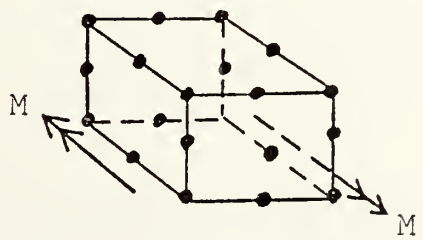


Figure 8c. 12-Node Brick Orientation for Case-3



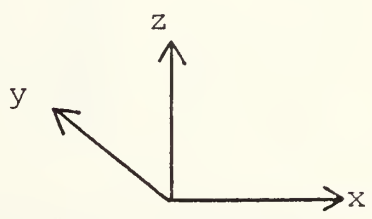


Figure 8d. 20-Node Brick; Case-4

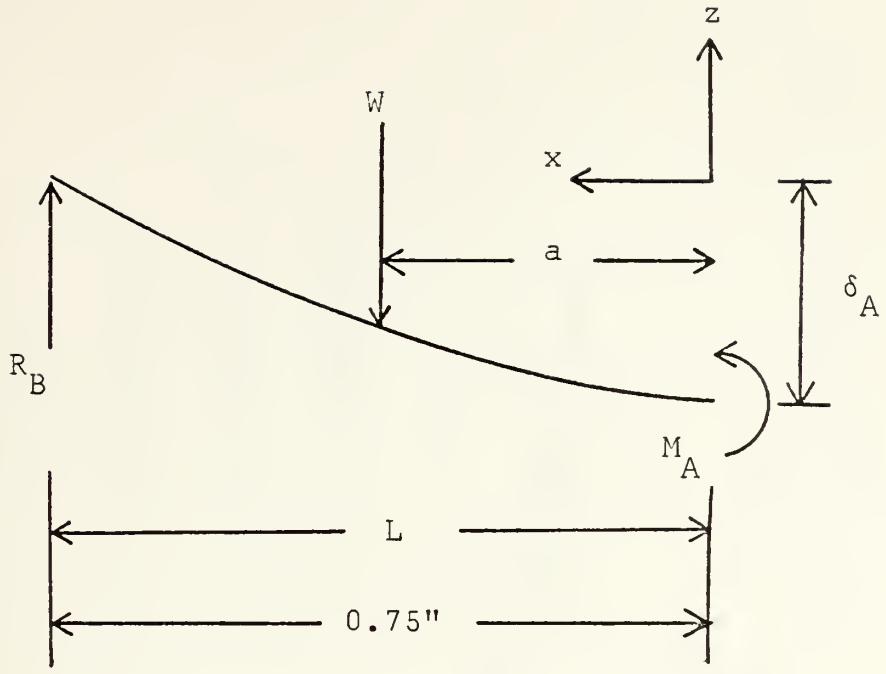


Figure 9a. Free Body Diagram for Flexure Case Study

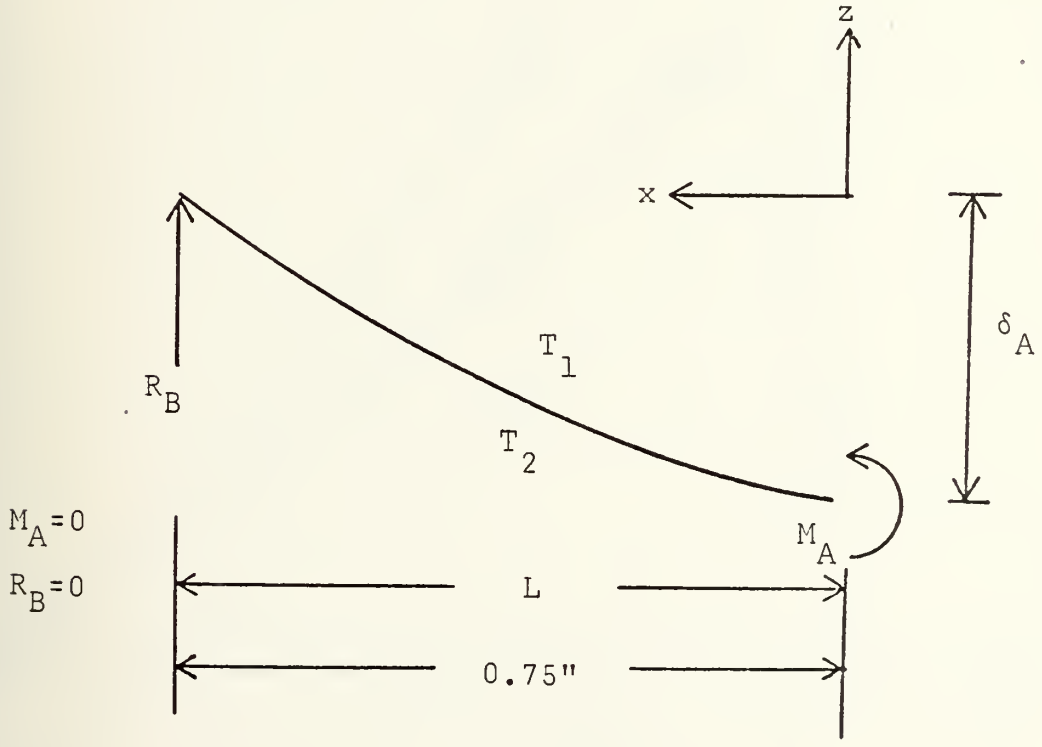


Figure 9b. Free Body Diagram for Temperature Gradient Case Study

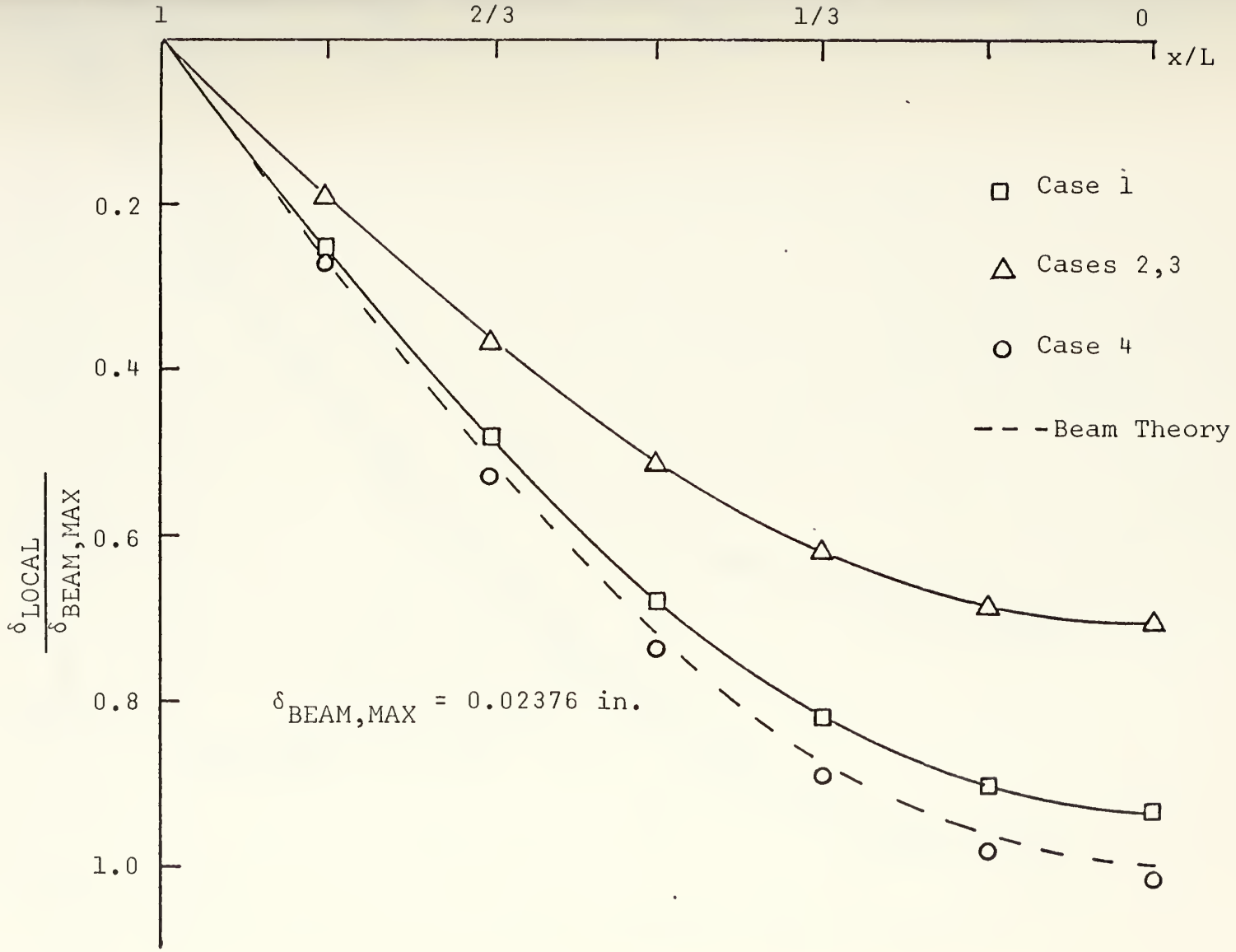


Figure 10. Normalized Displacement vs. Span for Steel Bend Specimen

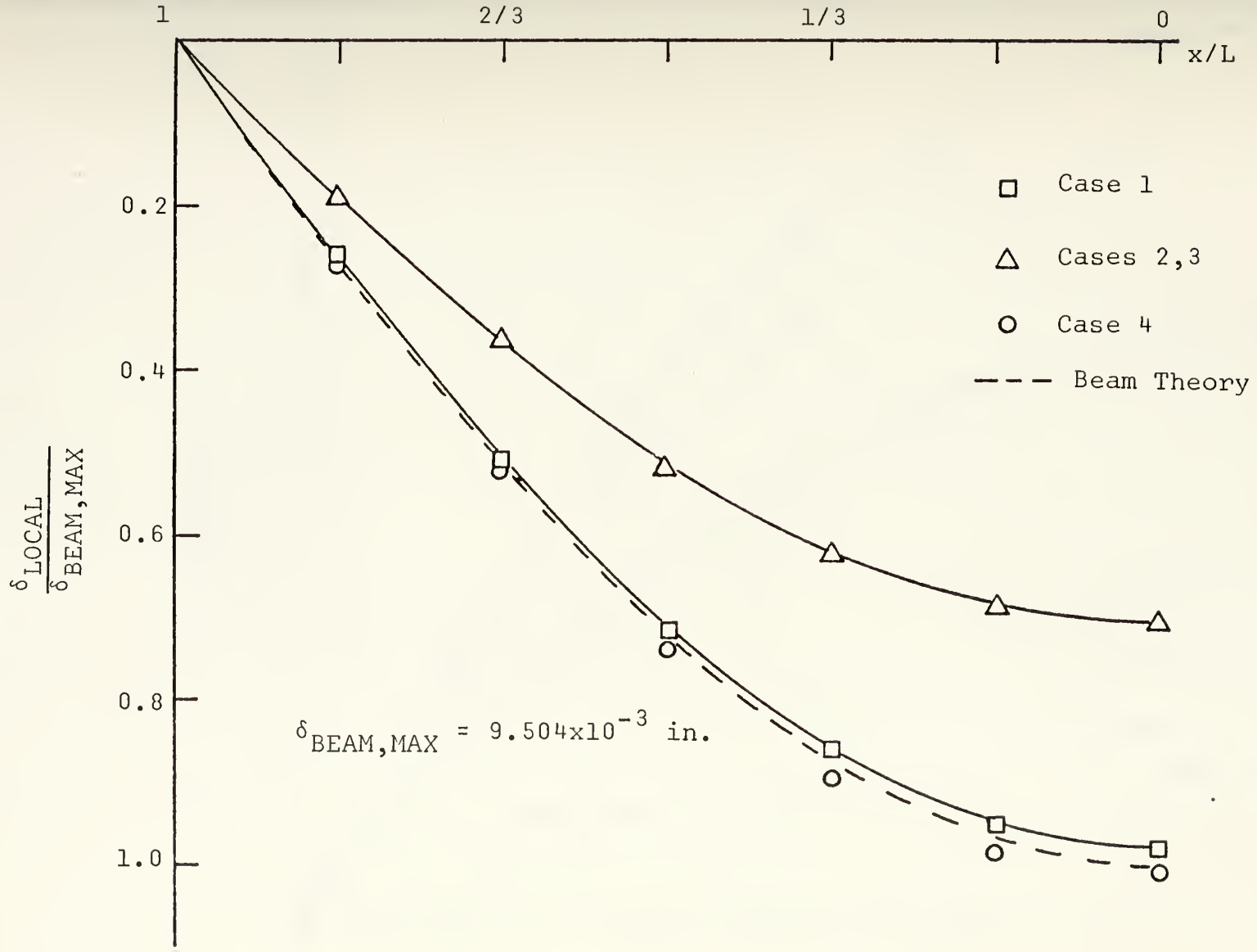


Figure 11. Normalized Displacement vs. Span for Ceramic Bend Specimen

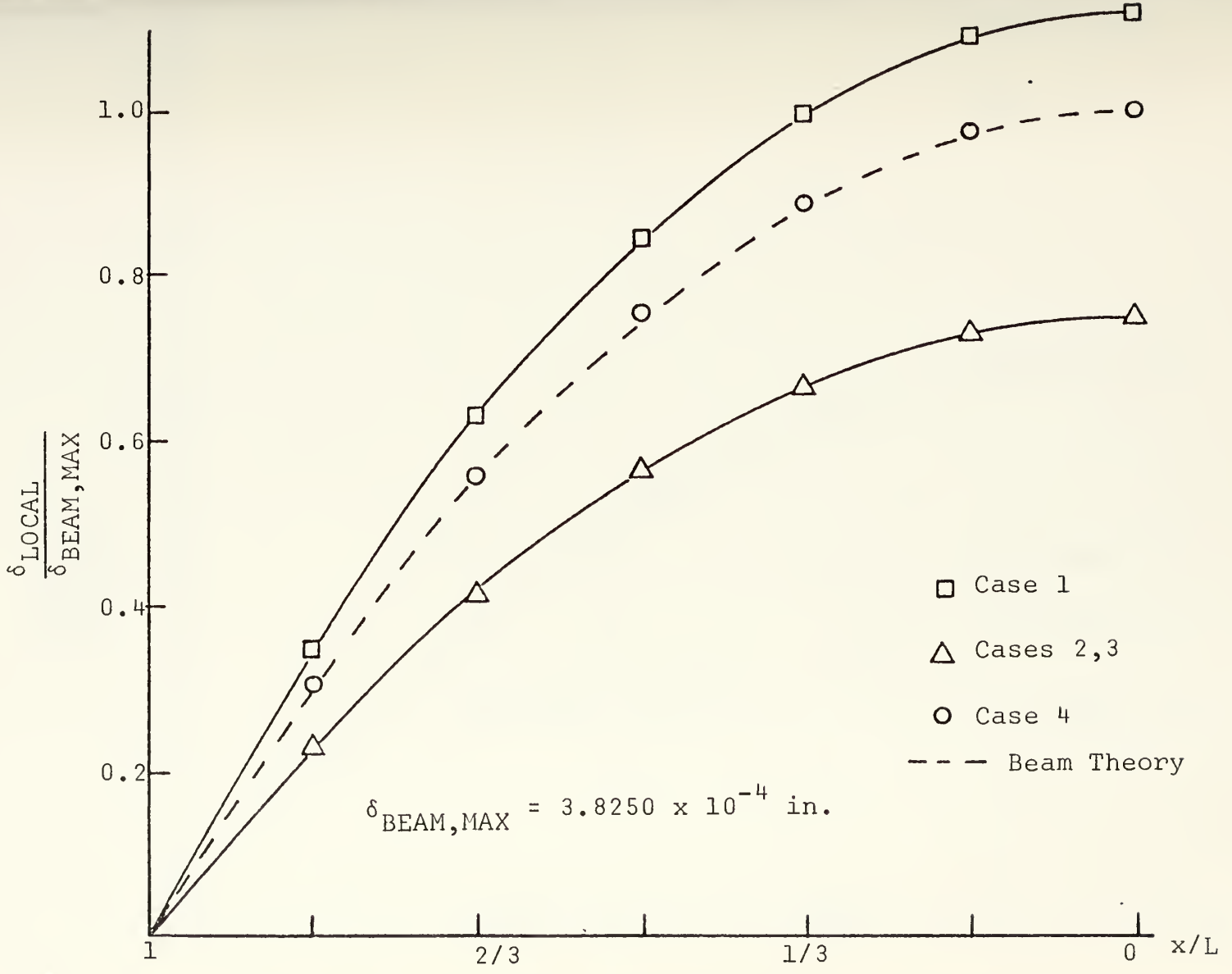


Figure 12. Normalized Displacement vs. Span for Temperature Gradient Analyses (ν=0.2)

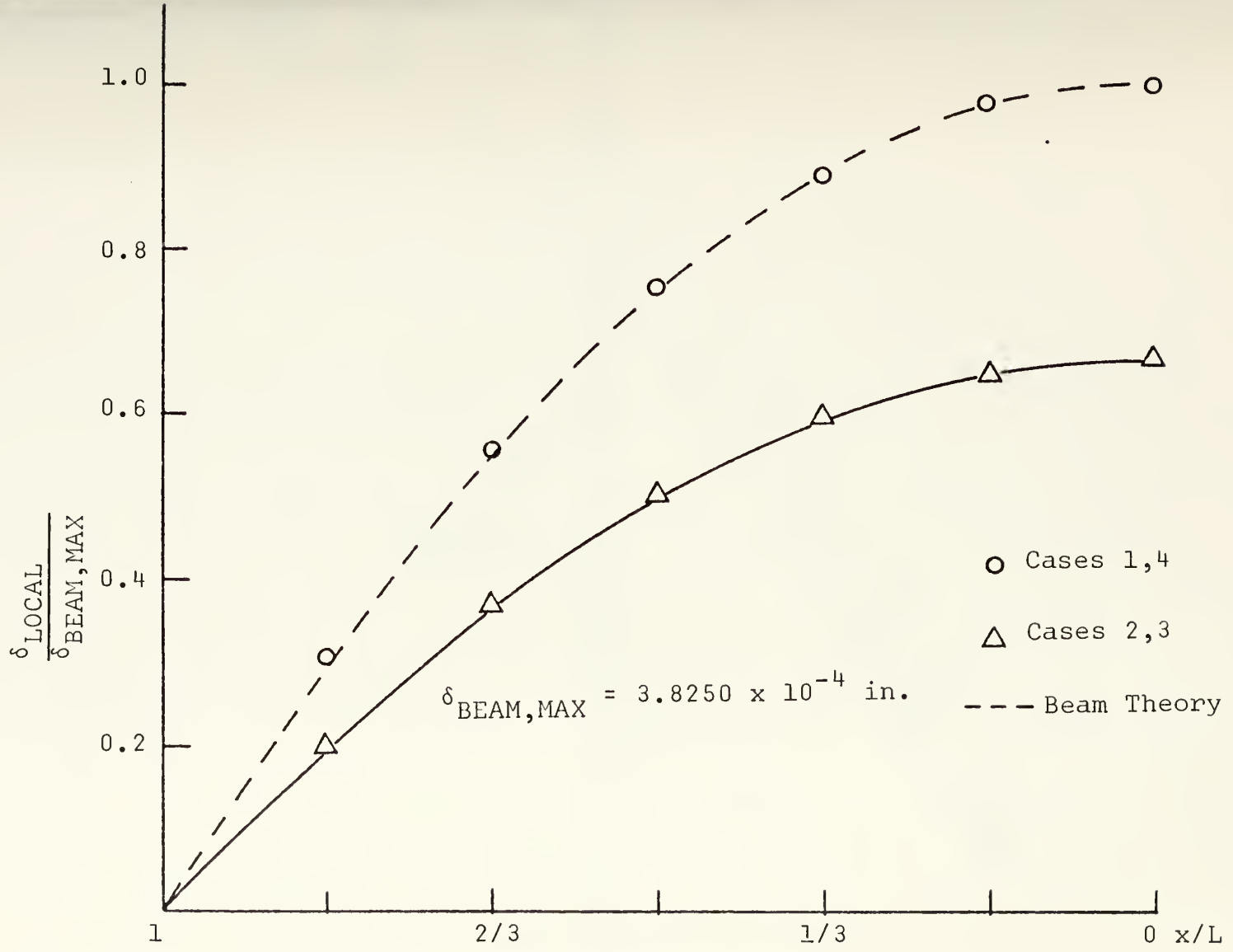


Figure 13. Normalized Displacement vs. Span for Temperature Gradient Analyses (ν=0)

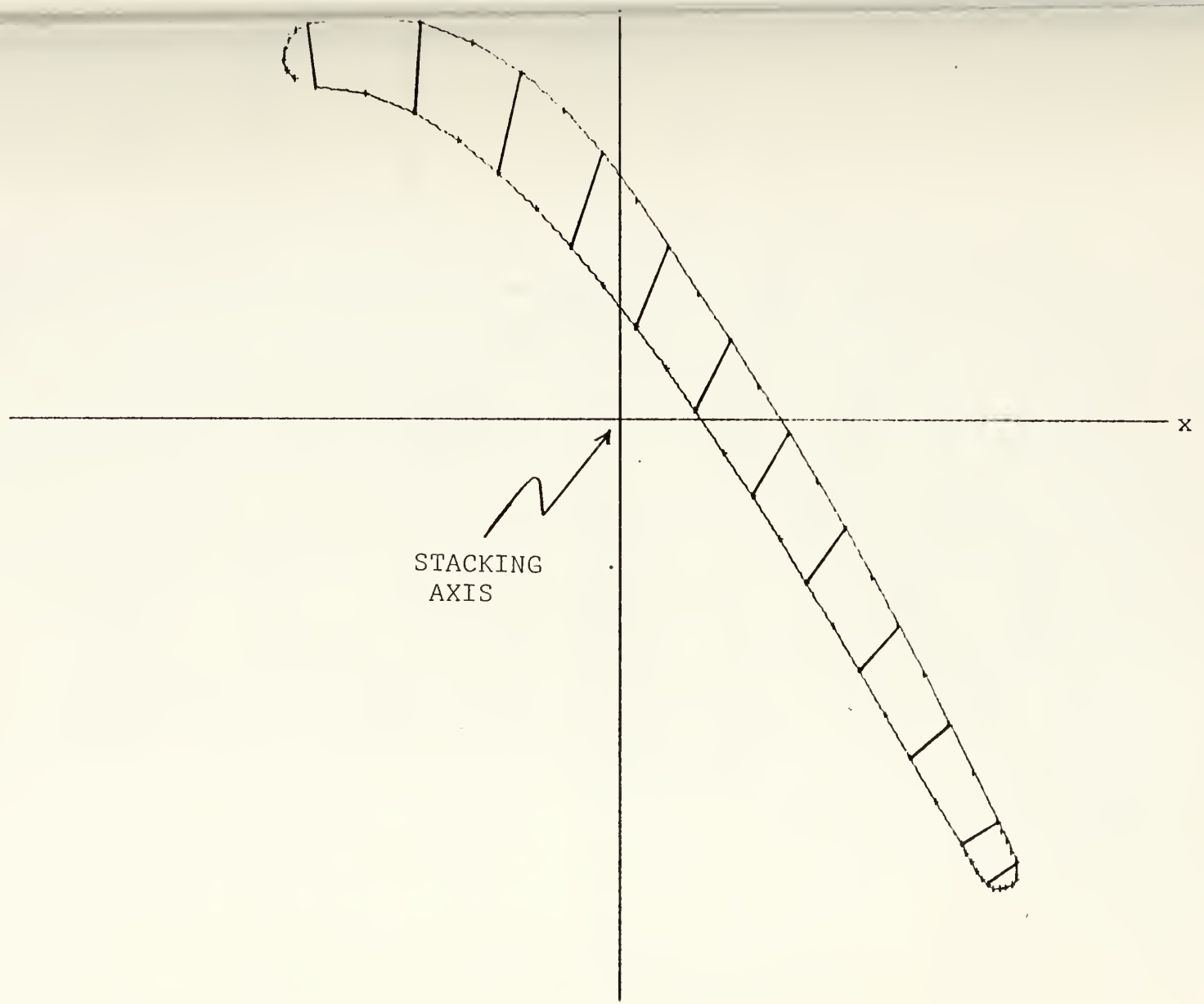


Figure 14. Airfoil Tip Section Profile

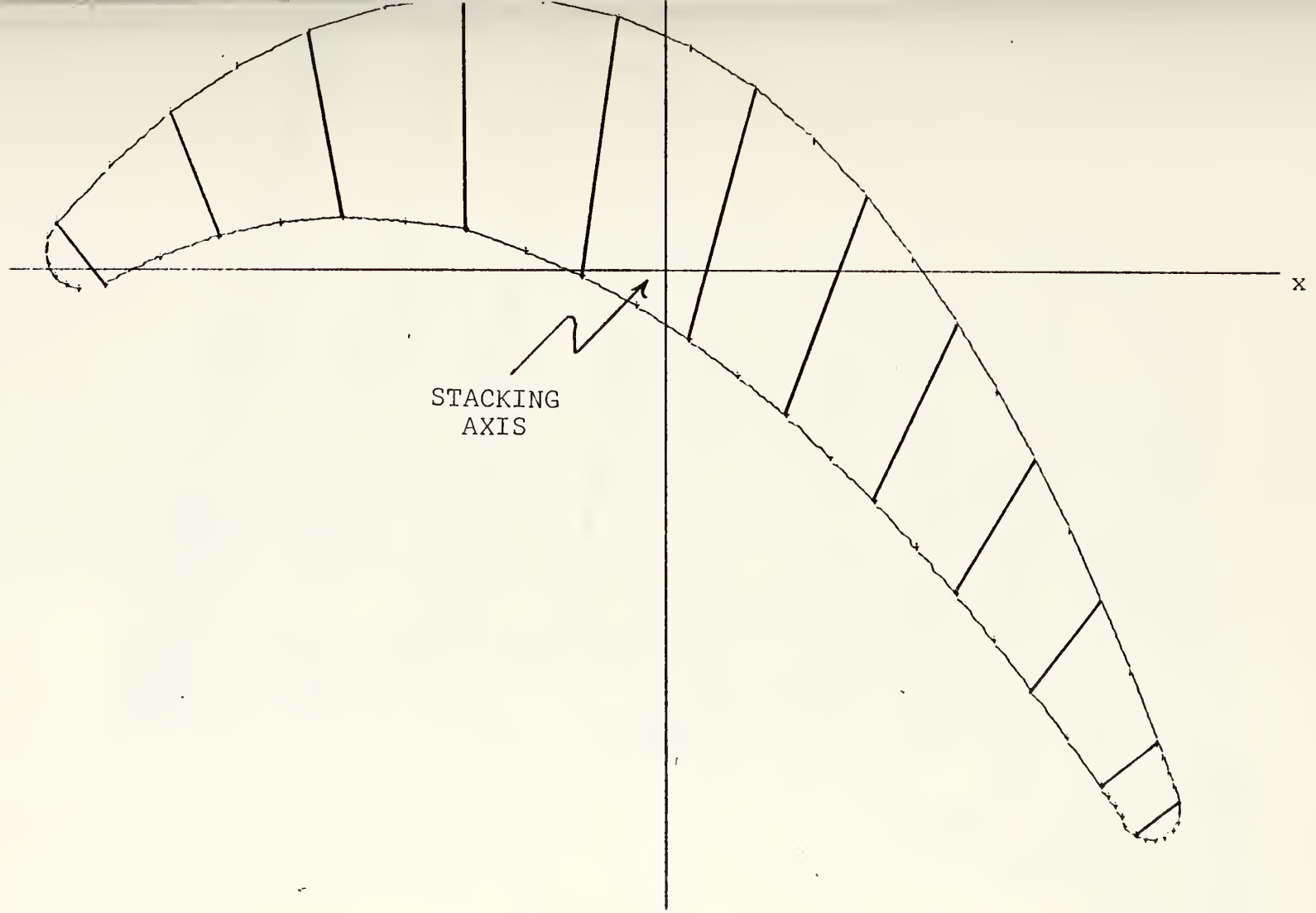


Figure 15. Airfoil Hub Section Profile

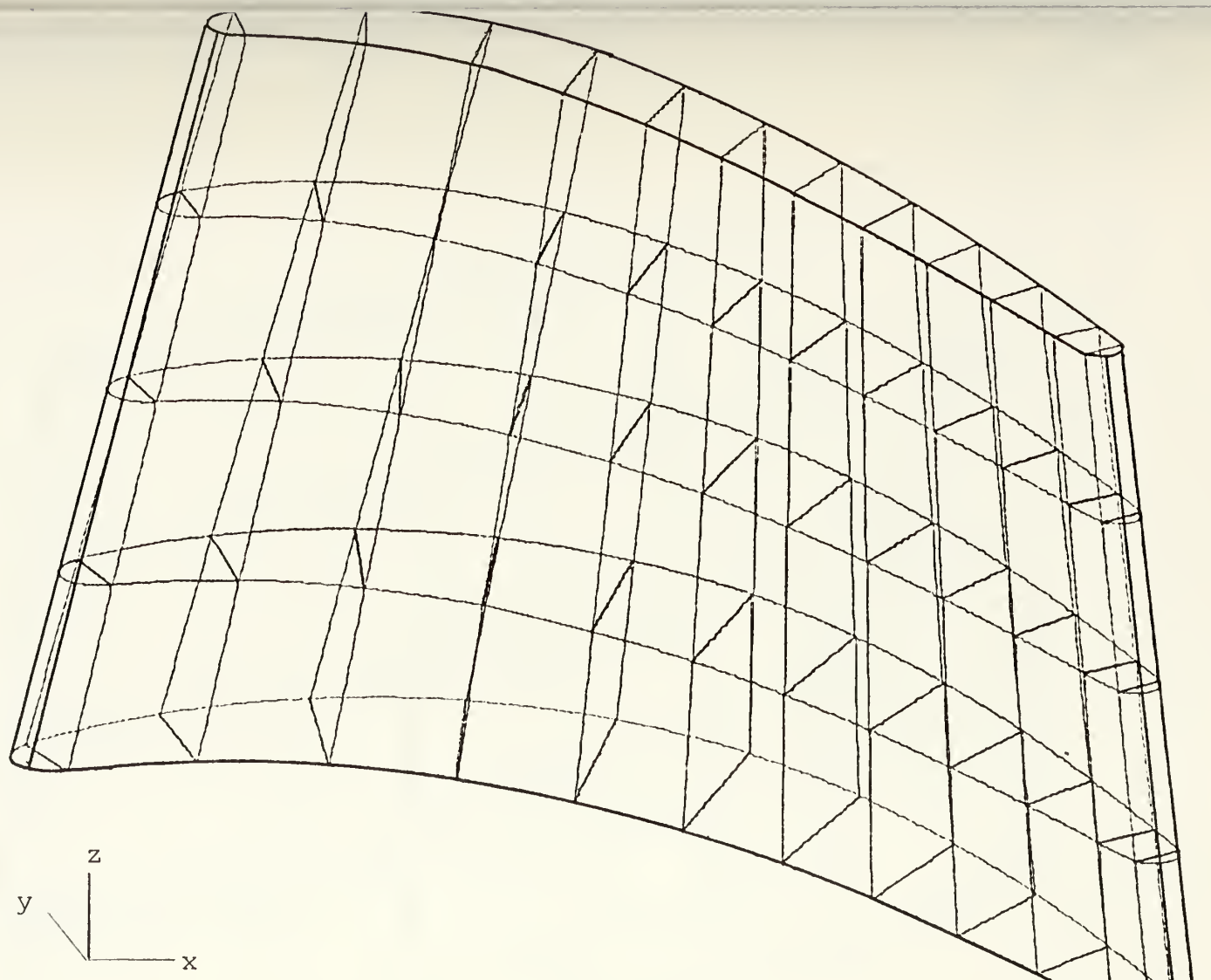


Figure 16. 20-Node Airfoil Mesh View 1

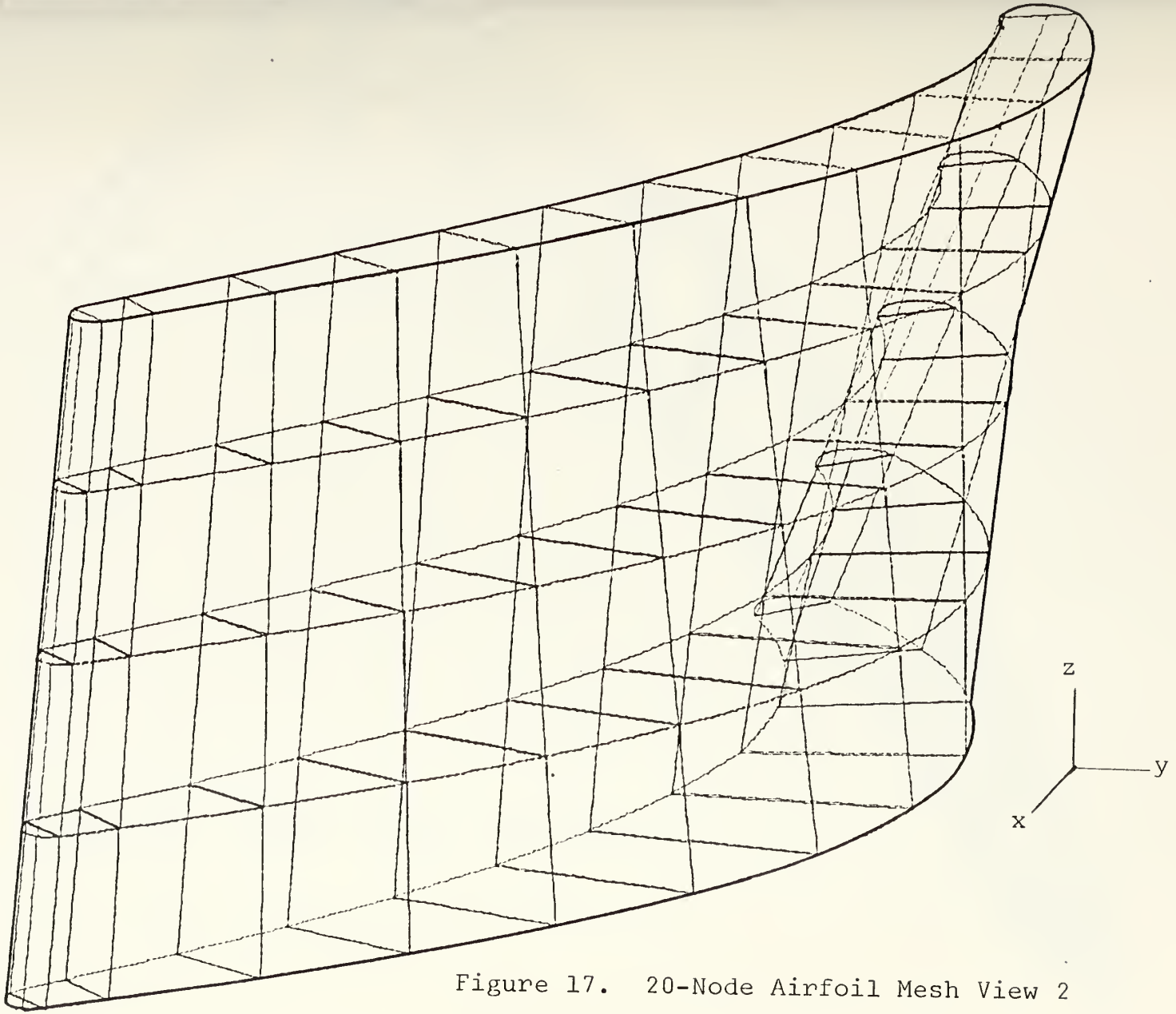


Figure 17. 20-Node Airfoil Mesh View 2

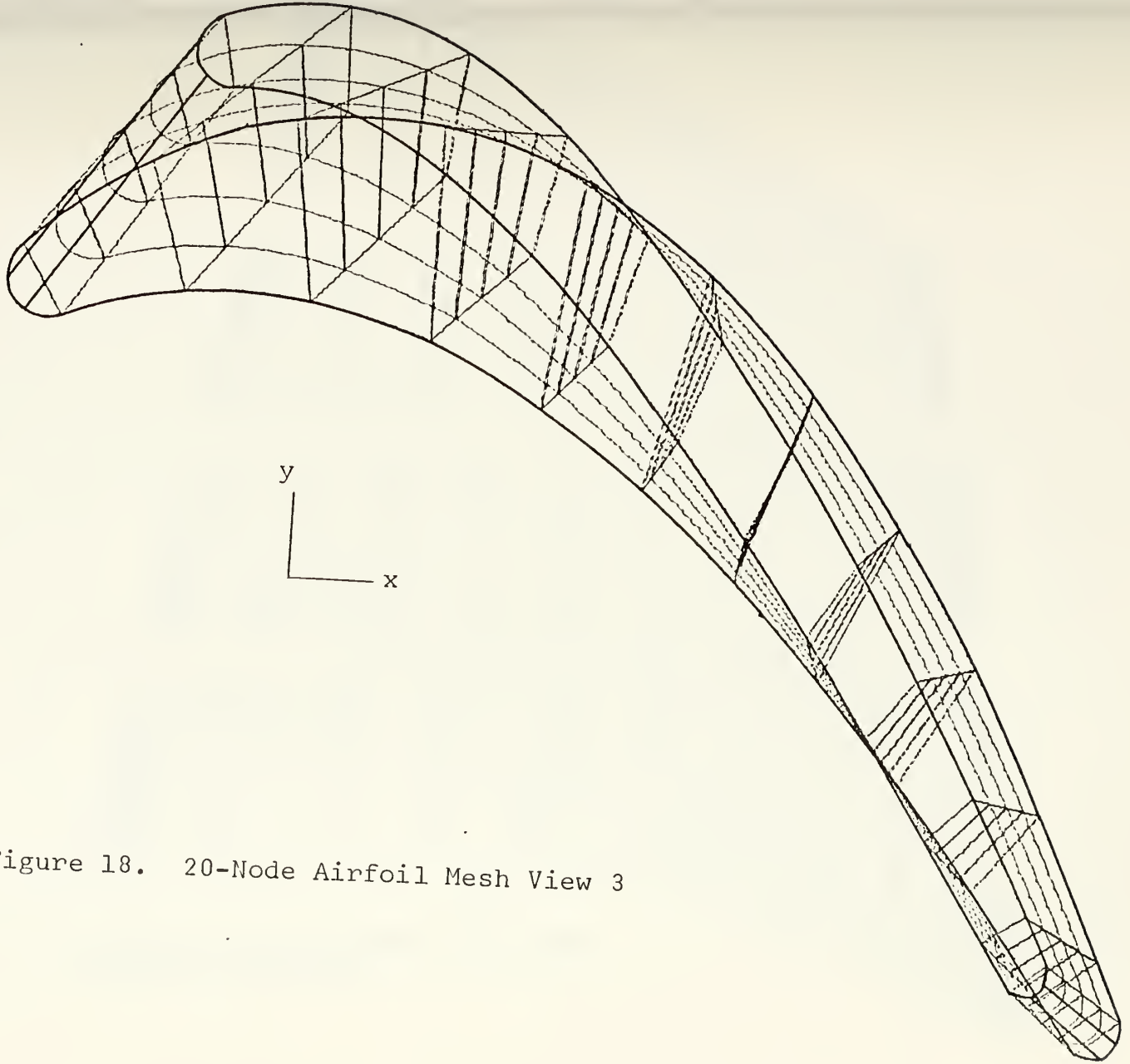


Figure 18. 20-Node Airfoil Mesh View 3

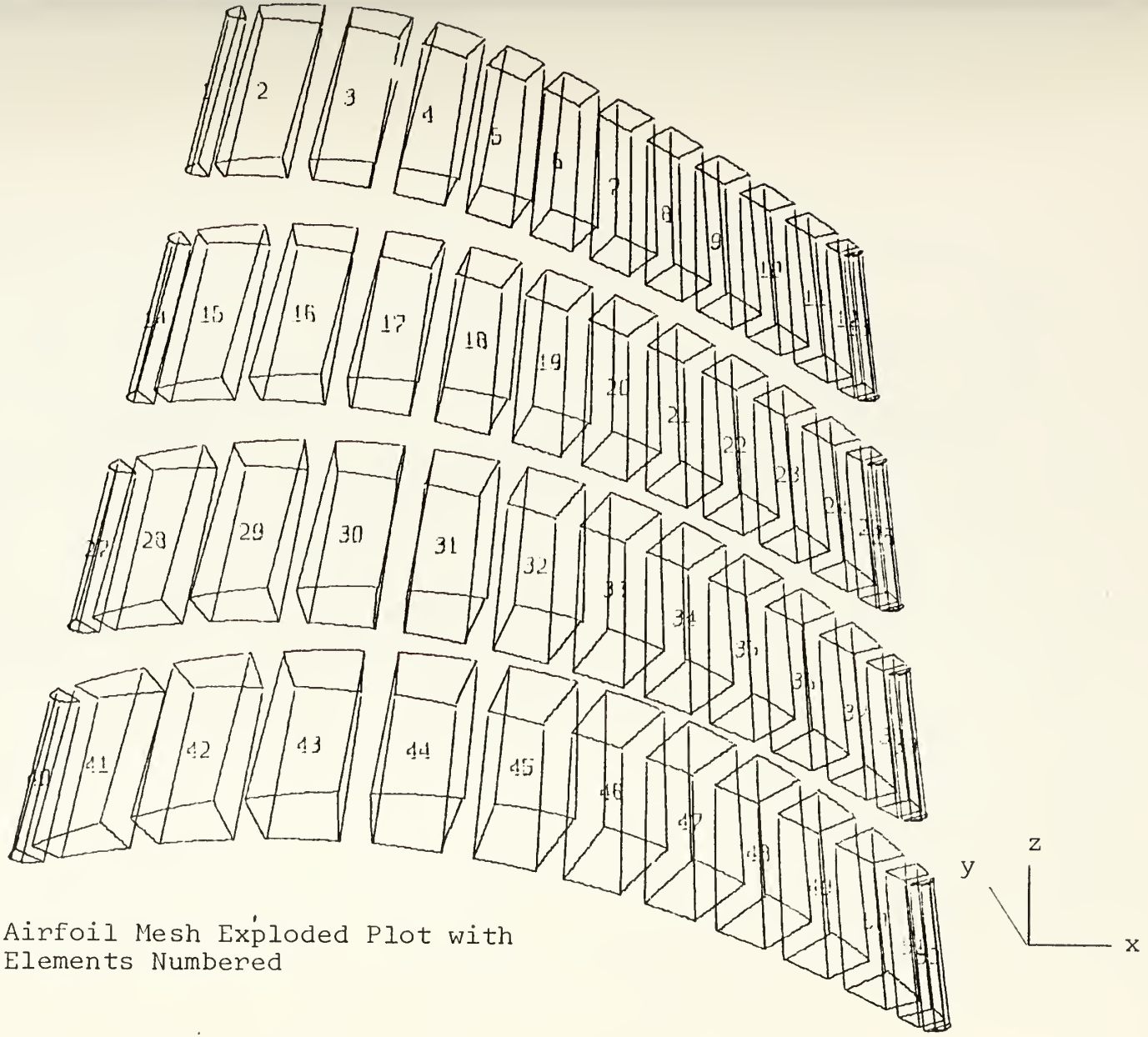


Figure 19. Airfoil Mesh Exploded Plot with Elements Numbered

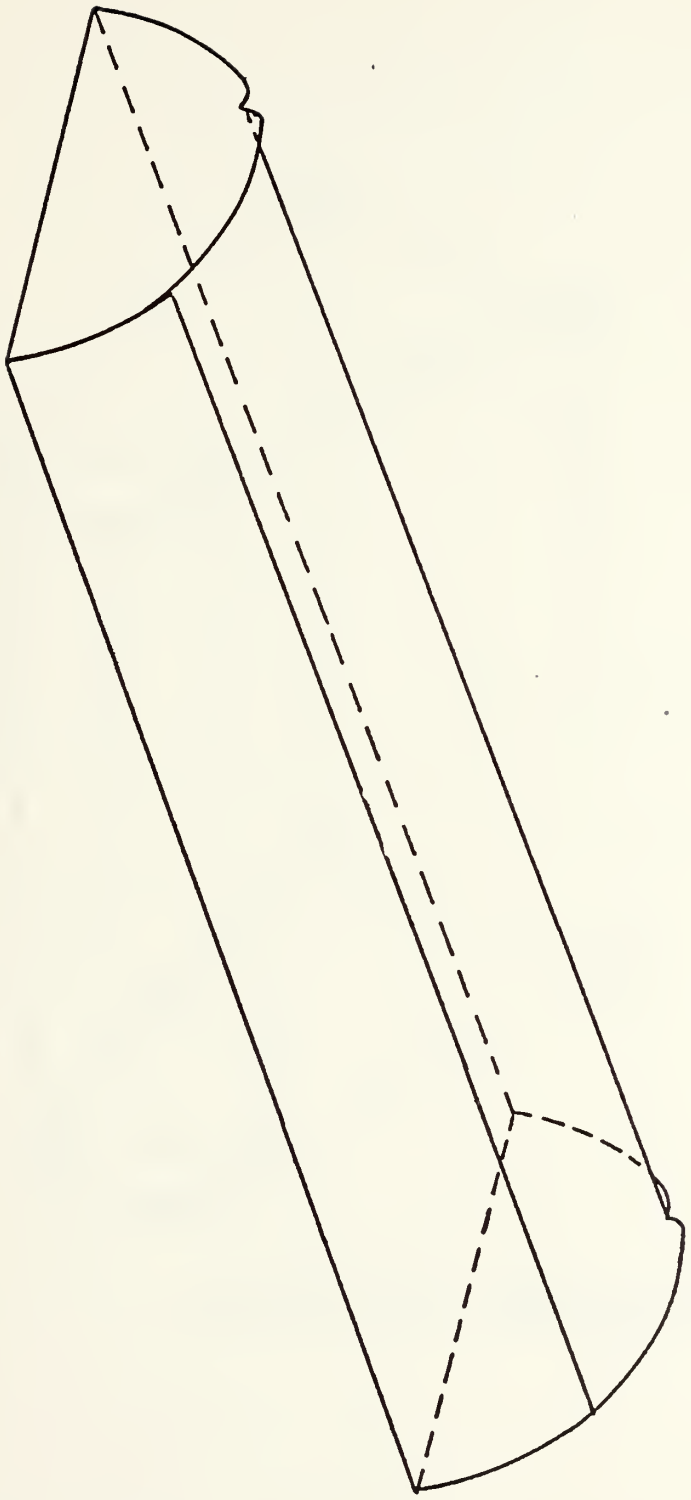


Figure 20. Trailing Edge Element Showing Re-entrant Angle

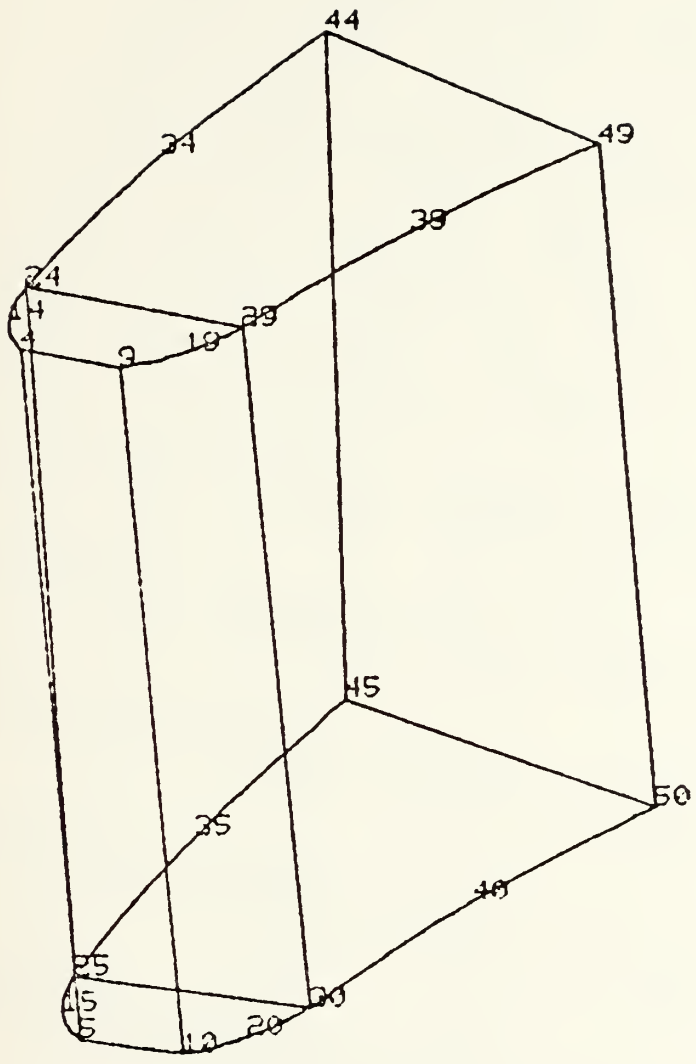


Figure 21. 12-Node Airfoil Section Showing Truncation of Leading Edge Element

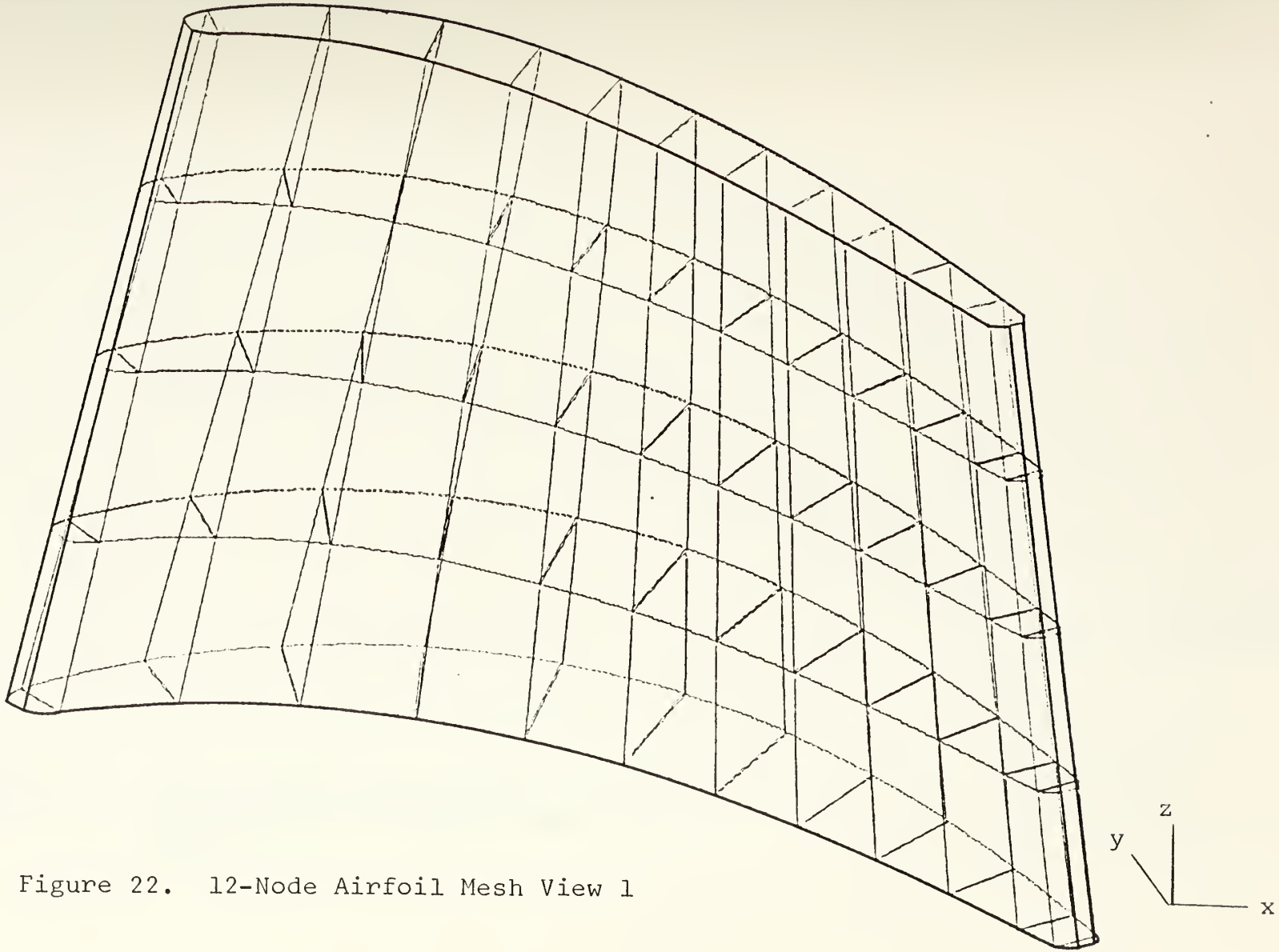


Figure 22. 12-Node Airfoil Mesh View 1

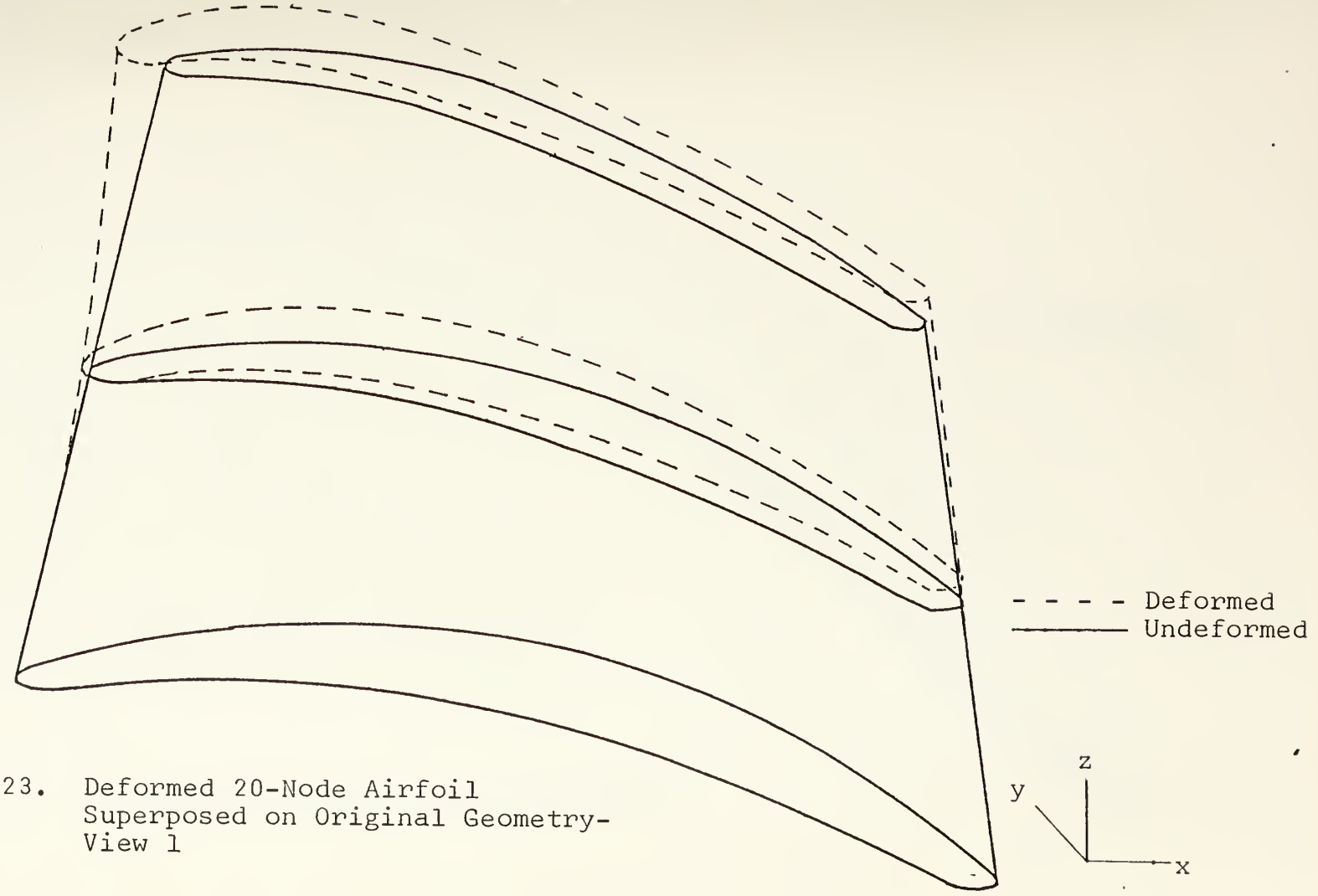


Figure 23. Deformed 20-Node Airfoil Superposed on Original Geometry-View 1

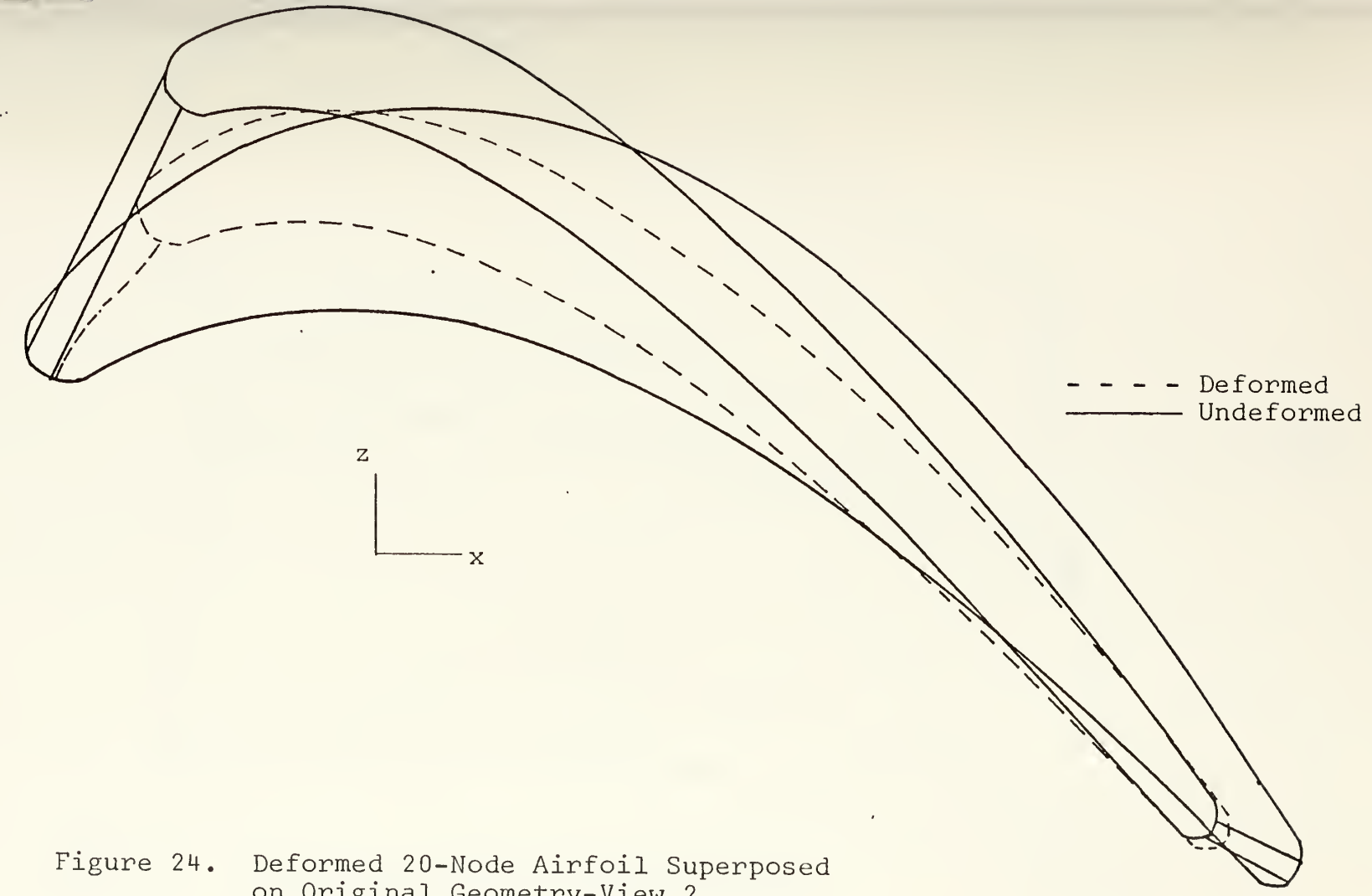


Figure 24. Deformed 20-Node Airfoil Superposed on Original Geometry-View 2

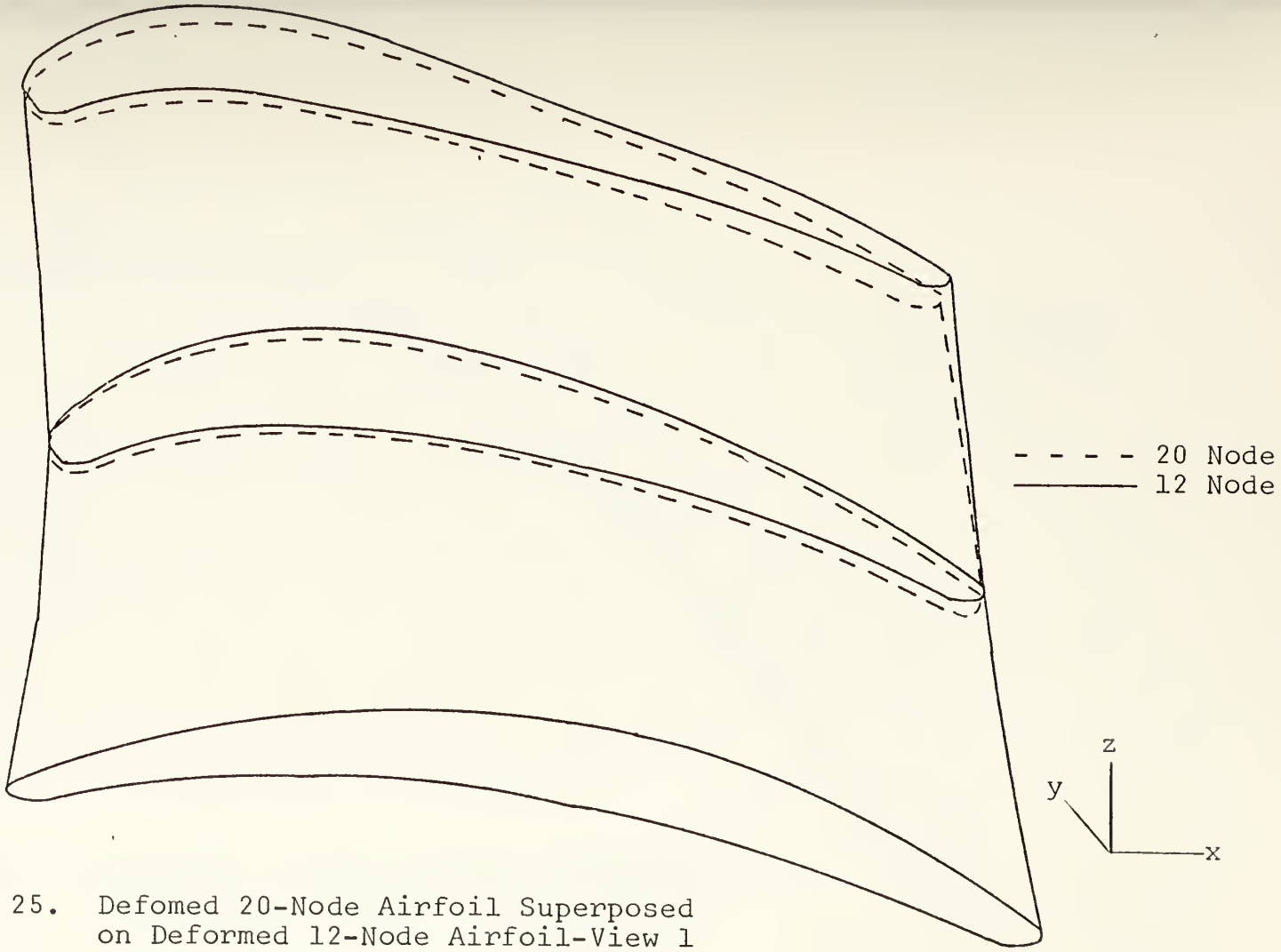


Figure 25. Defomed 20-Node Airfoil Superposed on Deformed 12-Node Airfoil-View 1

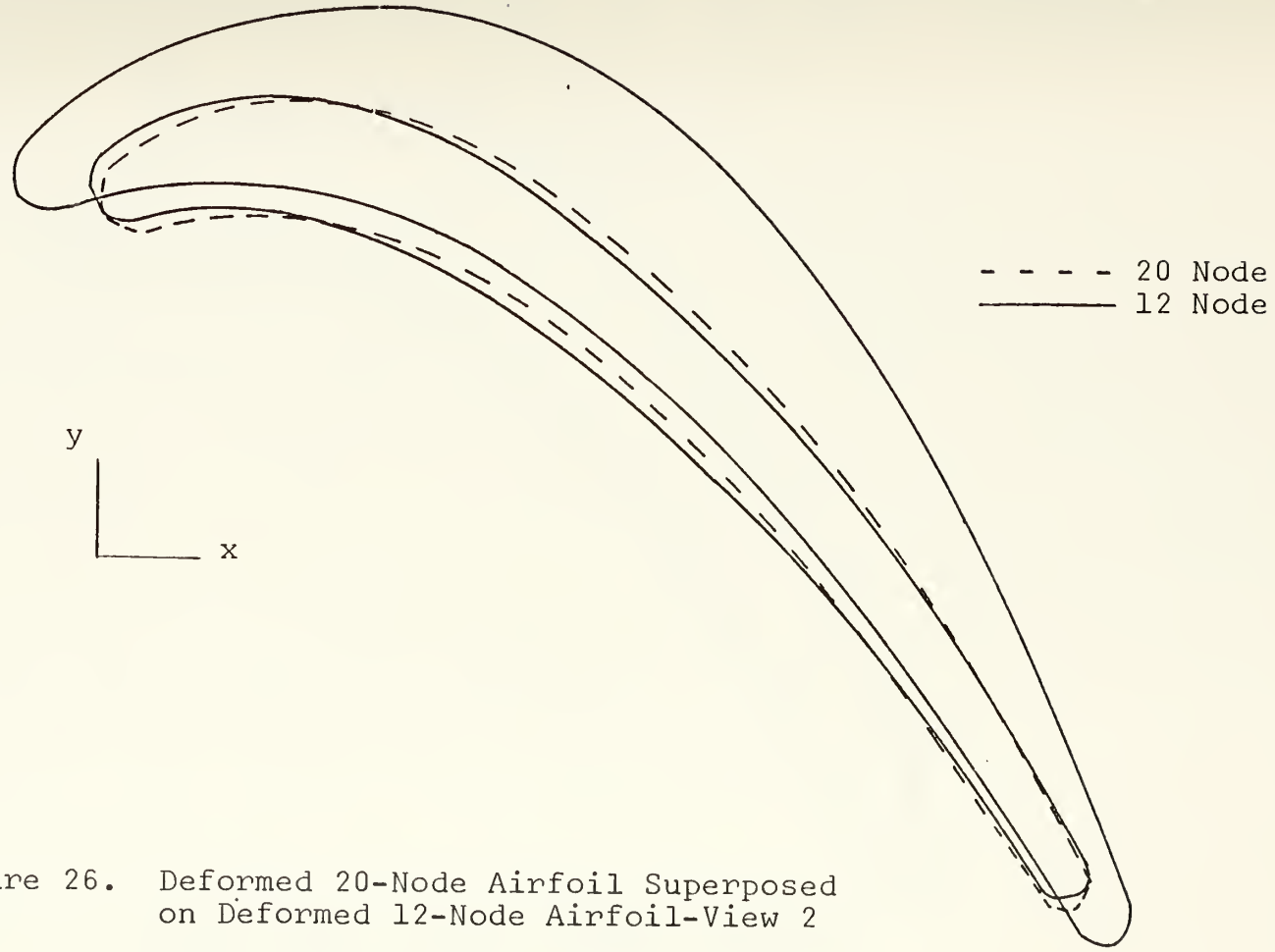


Figure 26. Deformed 20-Node Airfoil Superposed on Deformed 12-Node Airfoil-View 2

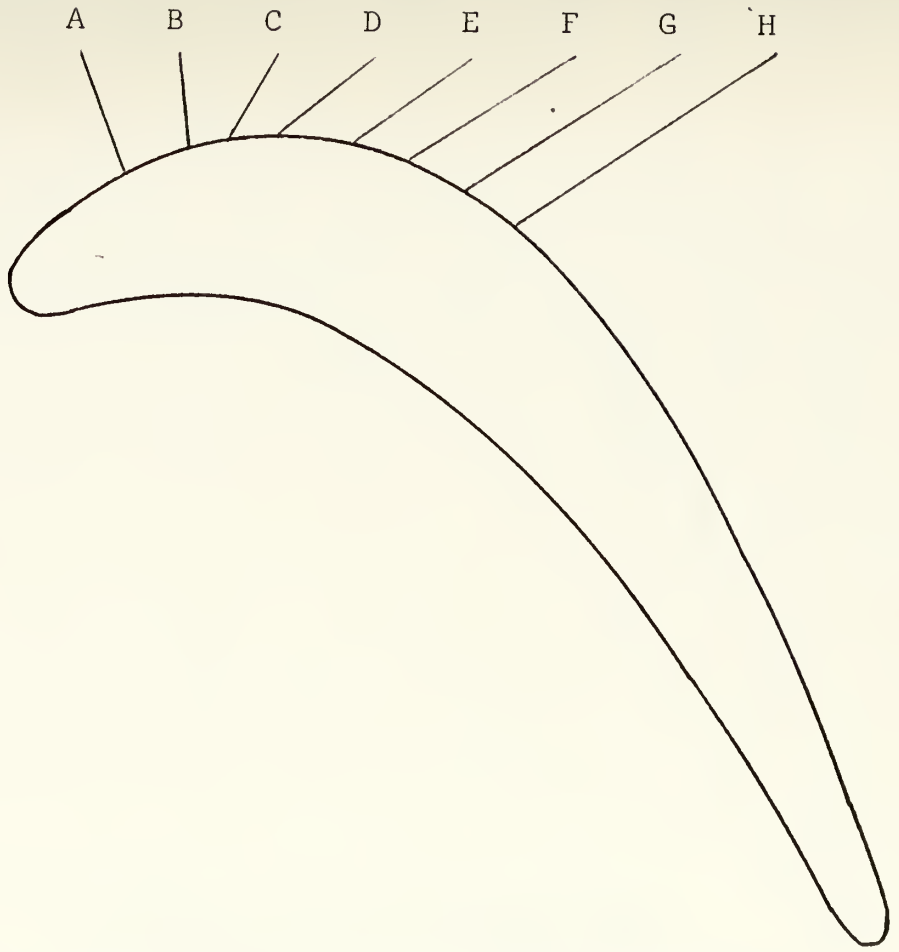


Figure 27. Airfoil Root Section Showing Location of Nodal Points Used for Comparison in Tables II and III

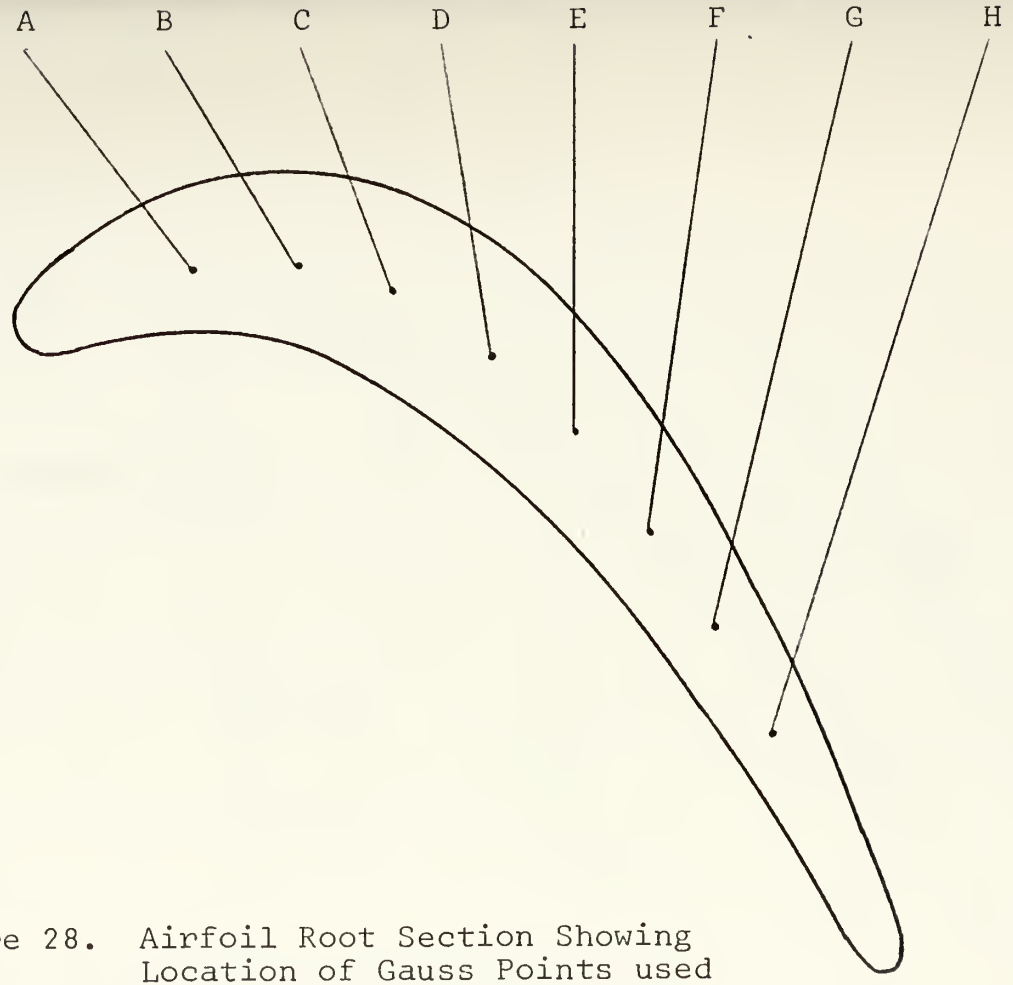


Figure 28. Airfoil Root Section Showing Location of Gauss Points used for Comparison in Table IV

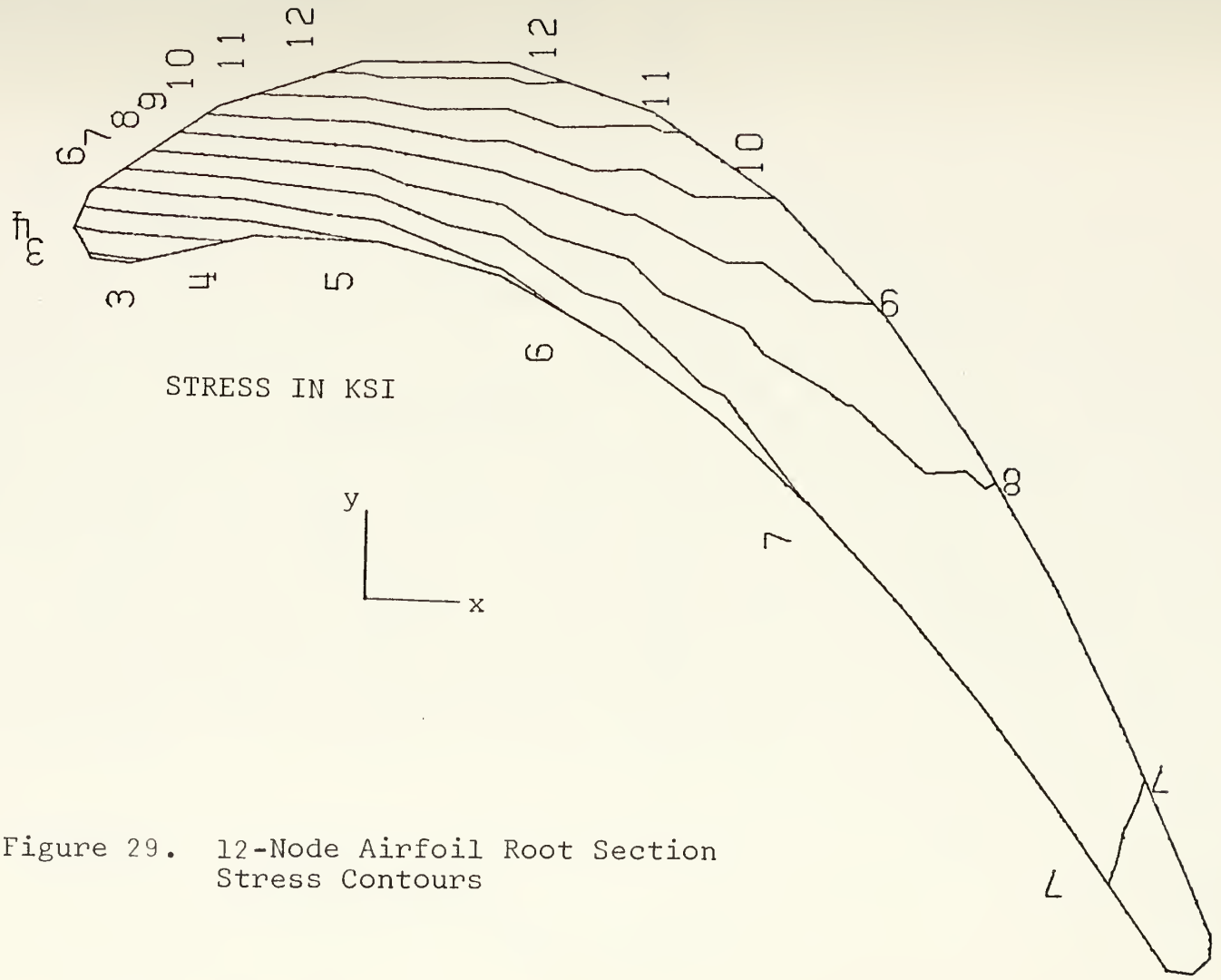


Figure 29. 12-Node Airfoil Root Section Stress Contours

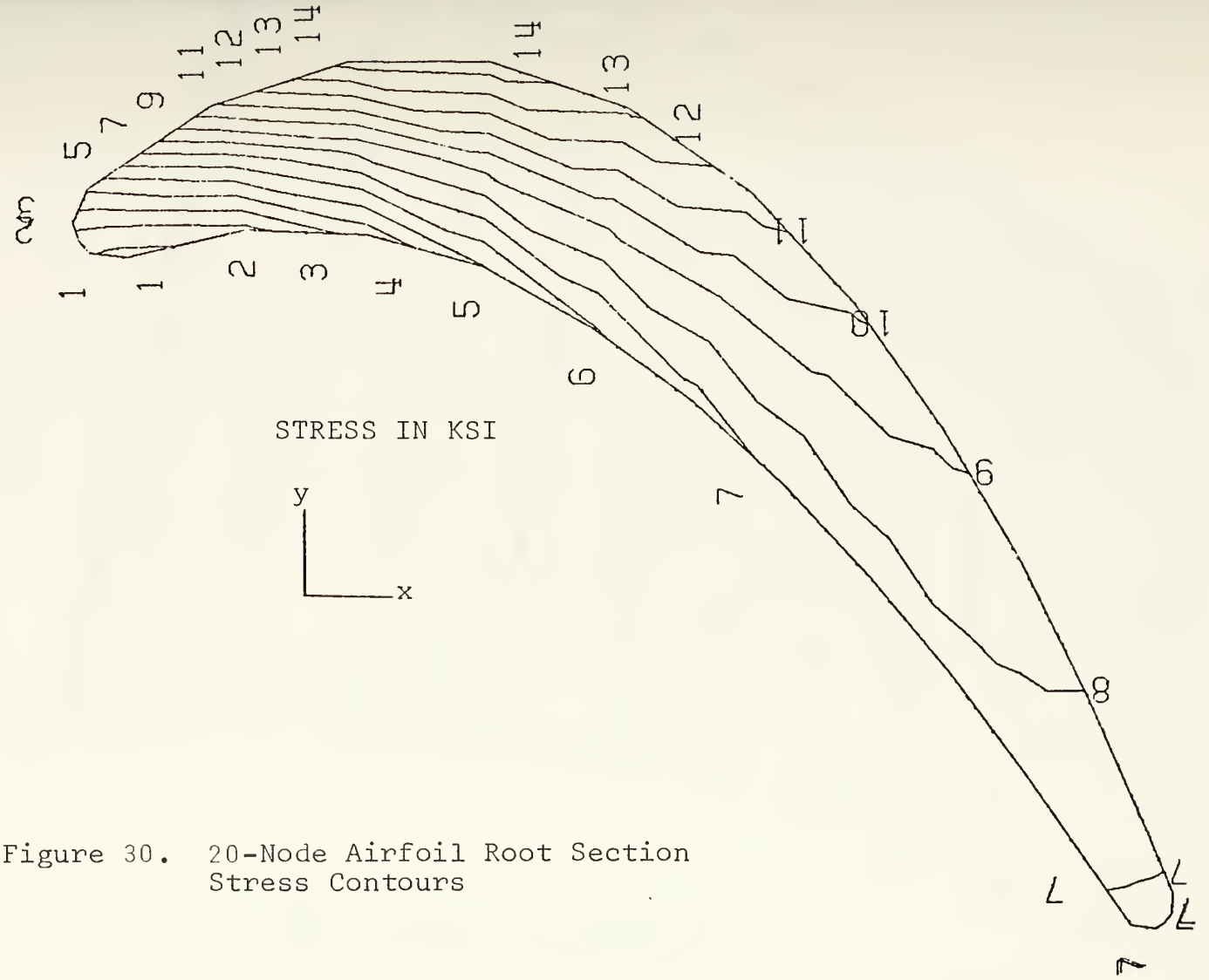


Figure 30. 20-Node Airfoil Root Section Stress Contours

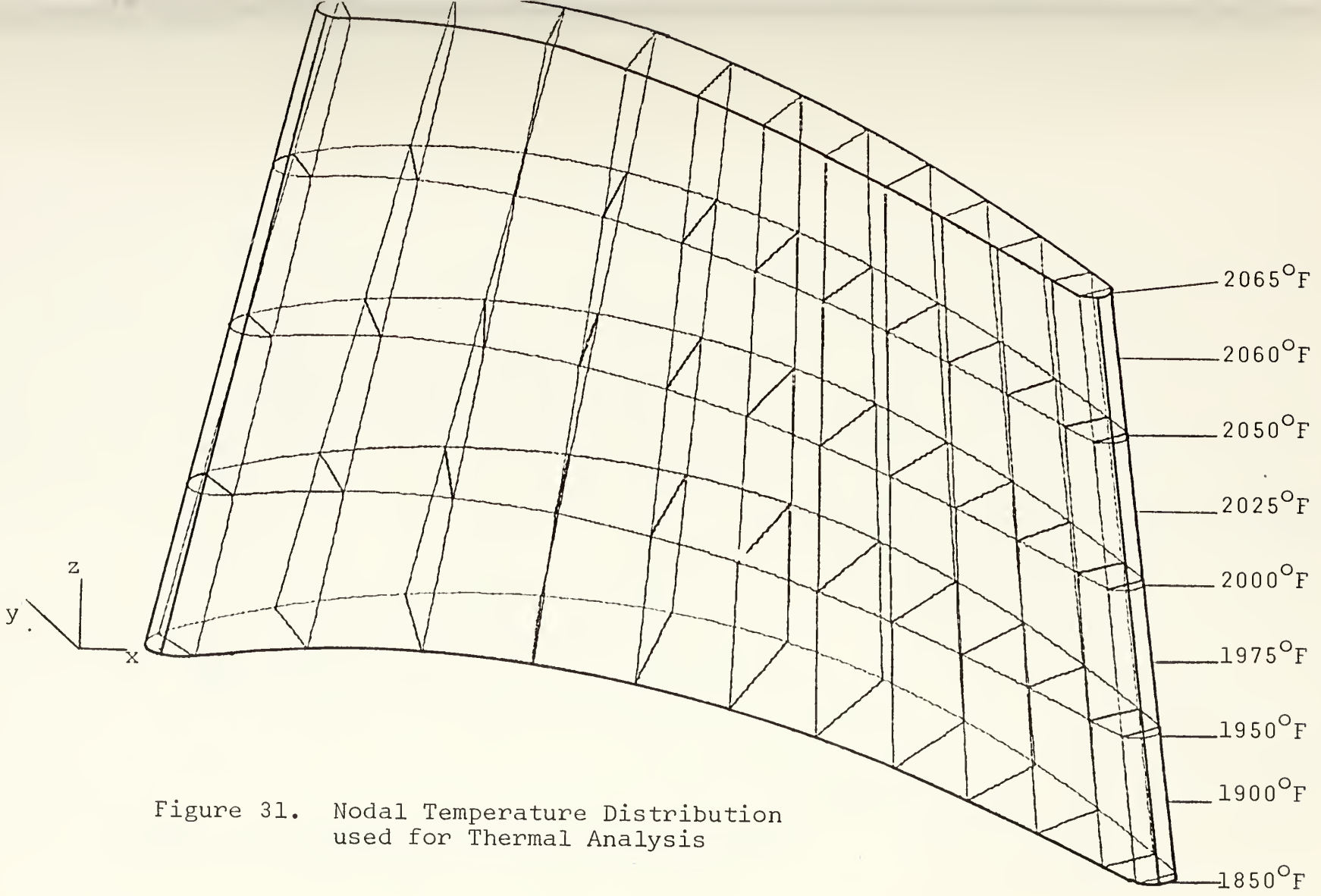


Figure 31. Nodal Temperature Distribution used for Thermal Analysis

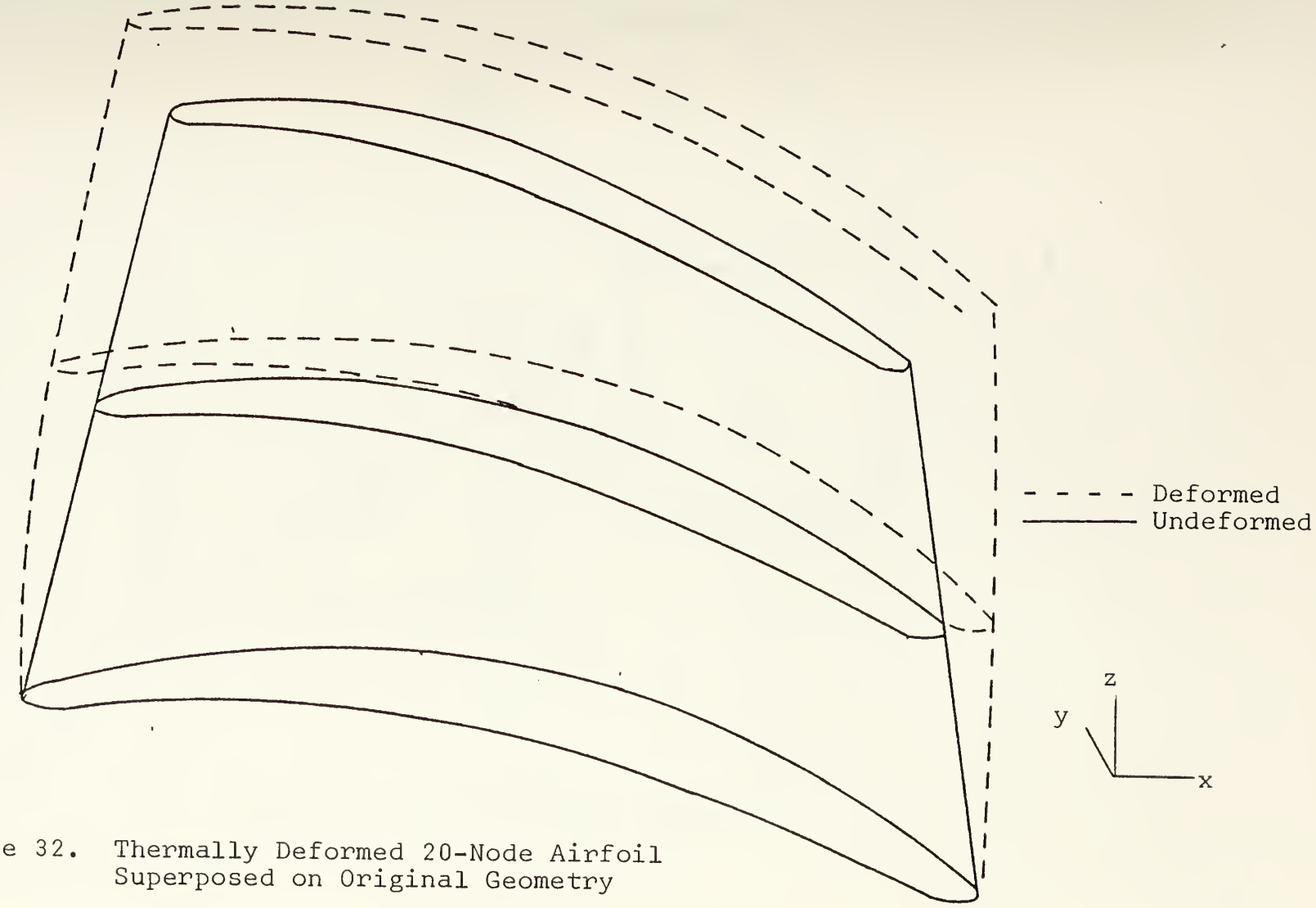


Figure 32. Thermally Deformed 20-Node Airfoil Superposed on Original Geometry

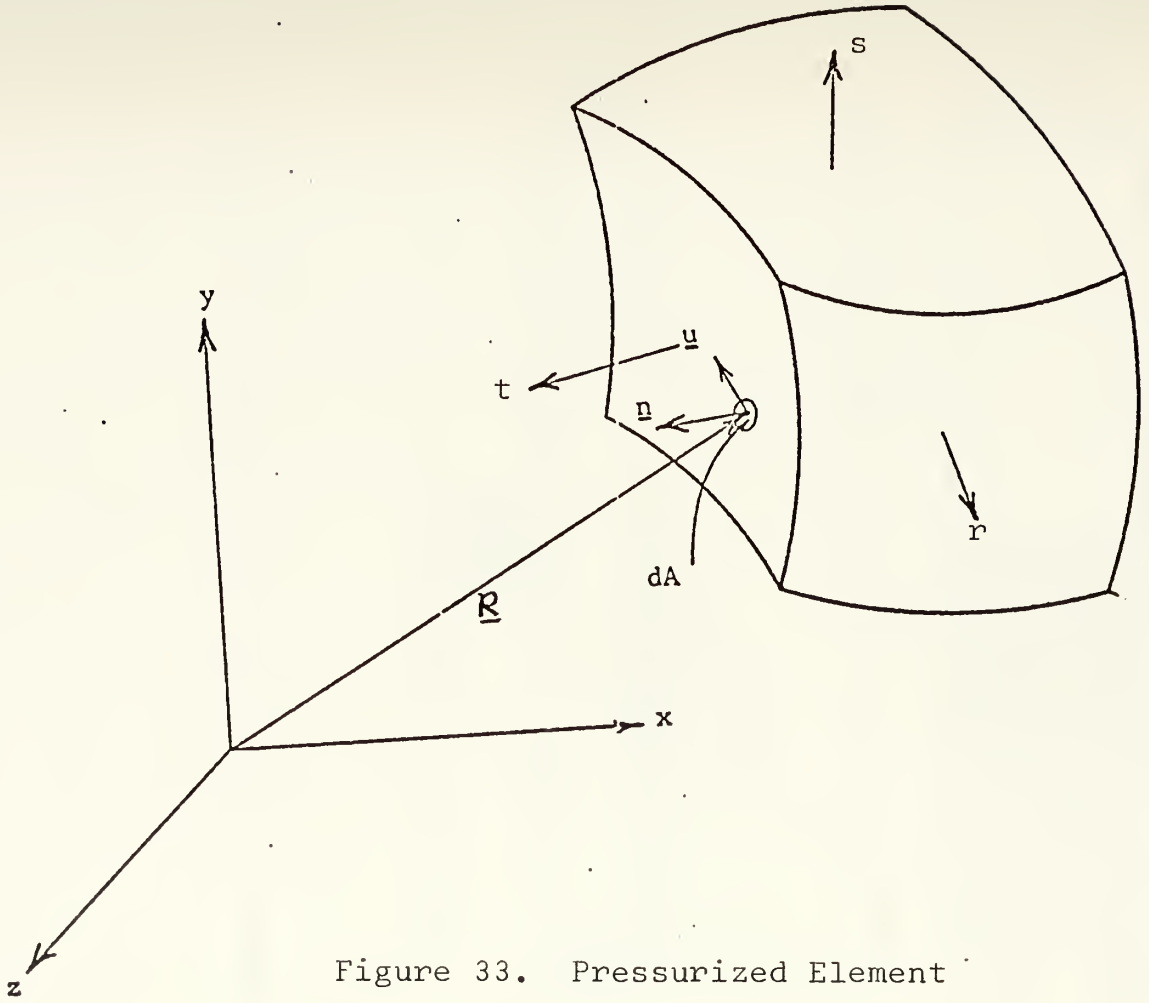


Figure 33. Pressurized Element

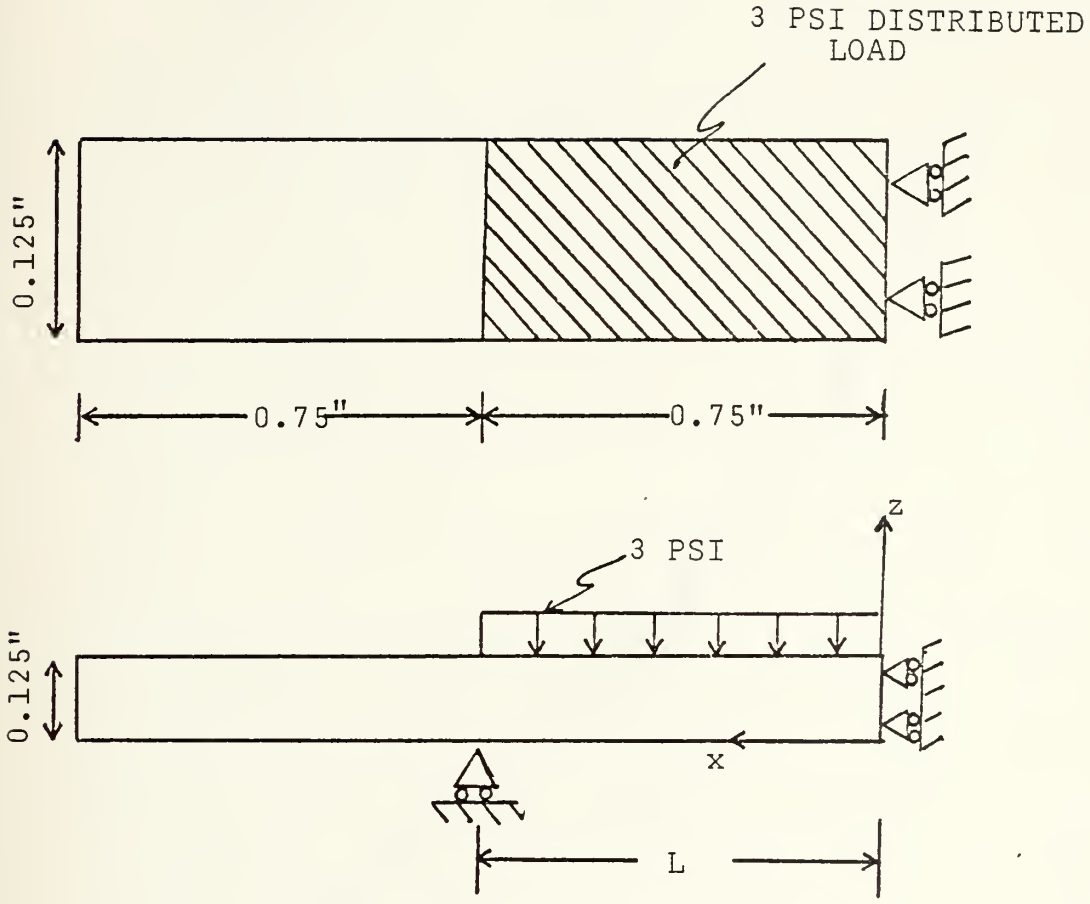


Figure 34. Pressure Load Verification Problem

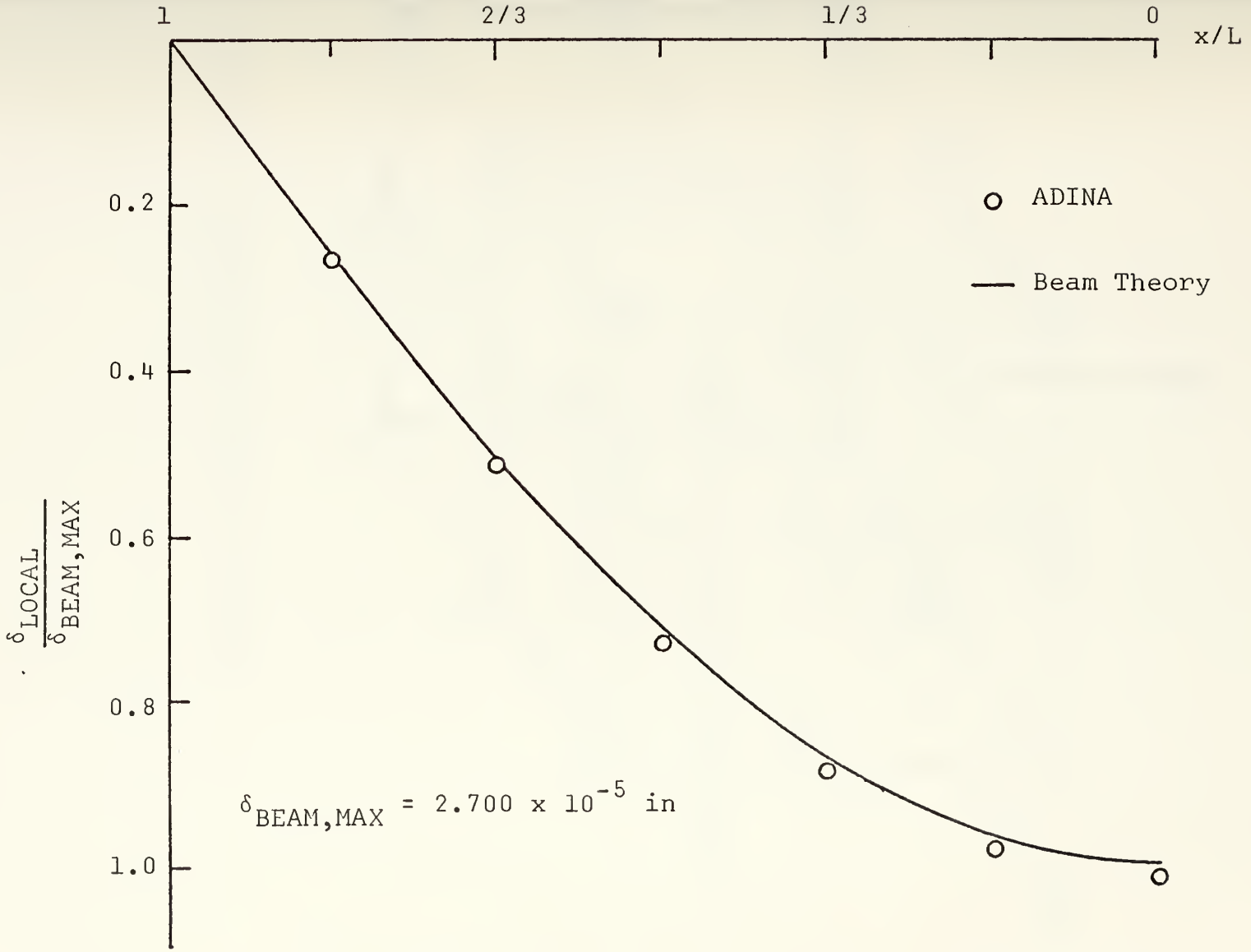


Figure 35. Normalized Displacement vs. Span for Bar Problem with Pressure Load

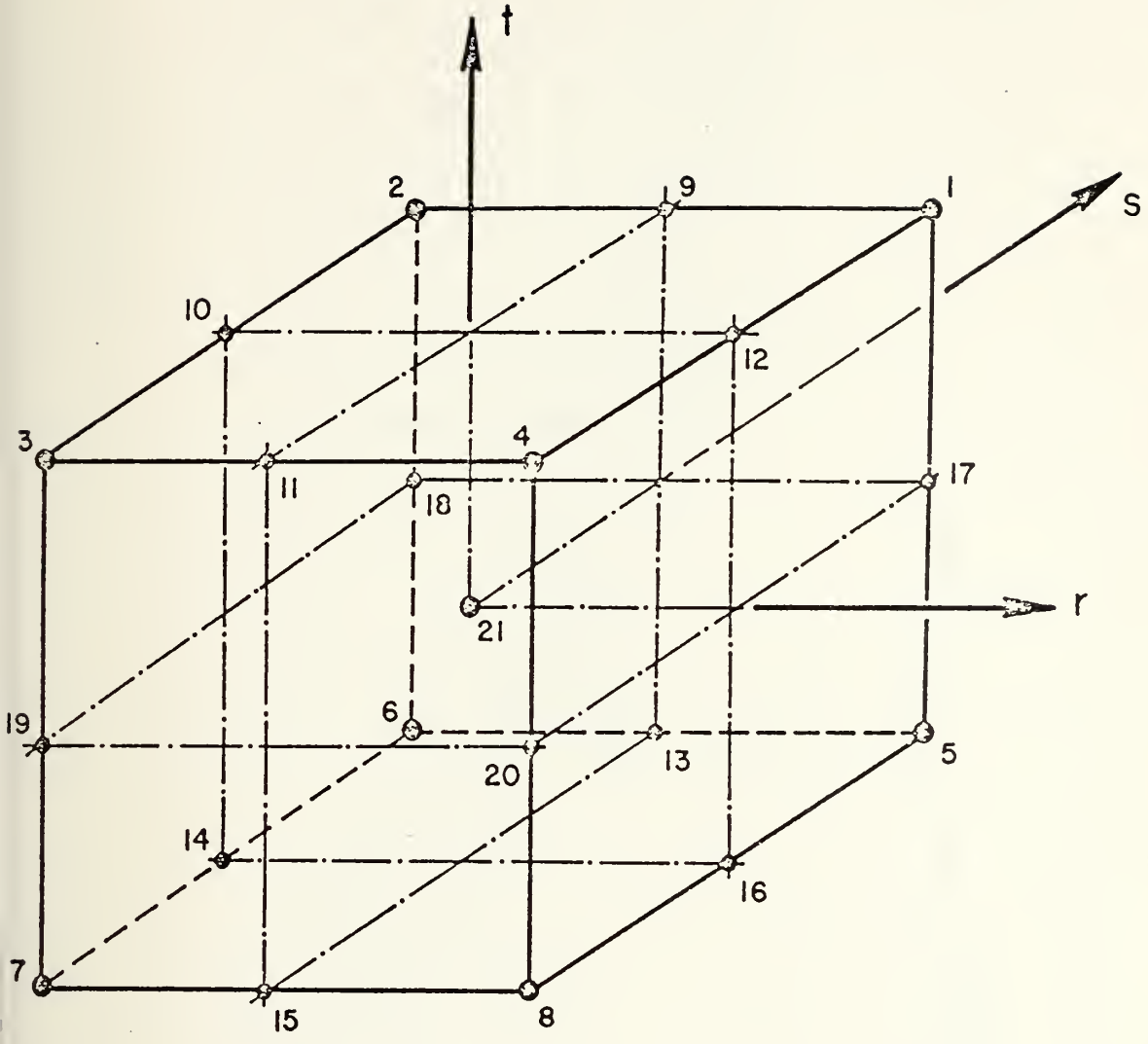


Figure 36. Hexahedral Element in Natural Coordinates

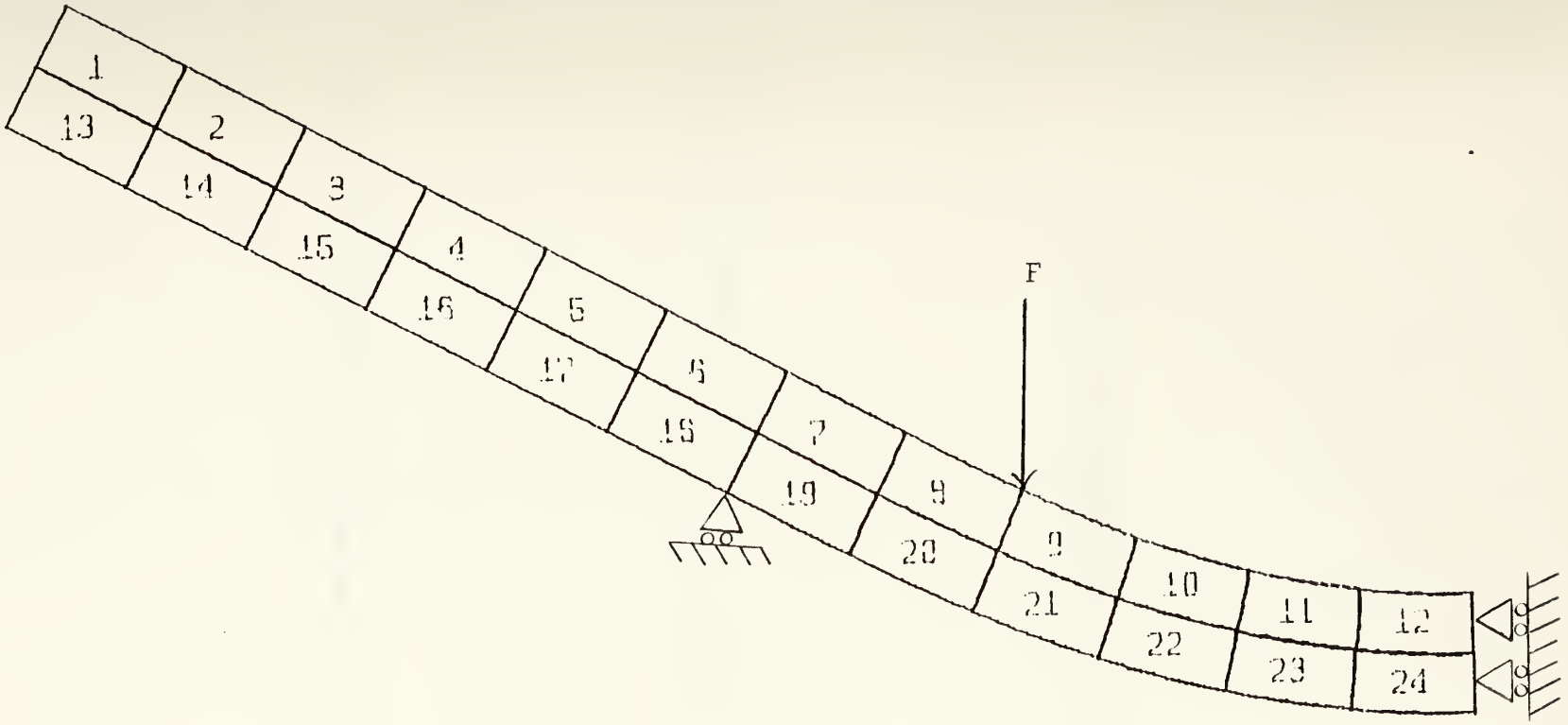


Figure 37. Deformed Structure for Flexure Problem

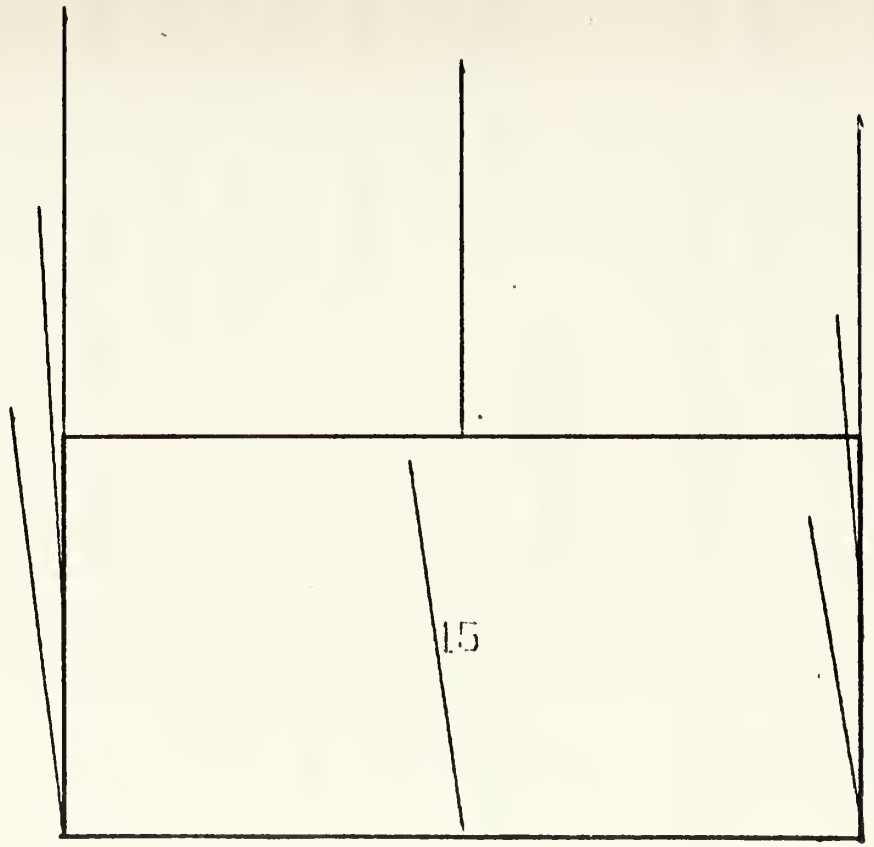


Figure 38. Postprocessing Using Displacement Vectors at the Nodes (Flexure Problem)

TABLE I: MESH CHARACTERISTICS

20-NODE BRICKS

Element type	20-node brick
Number of nodes	452
Number of elements	52
Degrees of freedom	1152
Mean half-bandwidth	93
Maximum half-bandwidth	129
Number of stiffness matrix elements	107064

12-NODE BRICKS

Element type	12-node brick
Number of nodes	270
Number of elements	52
Degrees of freedom	648
Mean half-bandwidth	45
Maximum half-bandwidth	66
Number of stiffness matrix elements	29106

TABLE II. COMPARISON OF MAXIMUM STRESSES IN AIRFOIL ROOT SECTION (Cf. Fig. 27)

LOCATION	FOIL	σ_{xx} (PSI)	σ_{yy} (PSI)	σ_{zz} (PSI)
A	12	2553	2553	10213
	20	2658	2658	10634
B	12	2897	2897	11588
	20	3211	3211	12843
C	12	3082	3082	12329
	20	3560	3560	14244
D	12	3130	3130	12520
	20	3656	3656	14626
E	12	3074	3074	12300
	20	3625	3625	15000
F	12	2946	2946	11784
	20	3447	3447	13788
G	12	2781	2781	11125
	20	3231	3231	12924
H	12	2608	2608	10432
	20	3000	3000	12001

TABLE III. COMPARISON OF MAXIMUM PRINCIPAL STRESSES IN AIRFOIL ROOT SECTION (cf. Fig. 27)

LOCATION	σ_1 12-NODE AIRFOIL (PSI)	σ_1 20-NODE AIRFOIL (PSI)	% DIFFERENCE
A	10441	10865	3.9
B	11790	12992	9.25
C	12519	14379	12.9
D	12706	14759	13.9
E	12486	14676	14.9
F	11969	13998	14.5
G	11308	13192	14.3
H	10608	12293	13.7

$$\sigma_1 > \sigma_2 > \sigma_3$$

$$\% \text{ Difference} \equiv \frac{\sigma_1(20) - \sigma_1(12)}{\sigma_1(20)}$$

TABLE IV. COMPARISON OF SELECTED THERMAL STRESSES IN AIRFOIL ROOT SECTION (cf. Fig. 28)

LOCATION	σ_{zz} 12 NODE (PSI)	σ_{zz} 20 NODE (PSI)	% DIFF
A	-11590	-11407	-1.6
B	-7744	-8664	10.6
C	-4387	-5759	23.8
D	-1813	-3487	48.0
E	-364	-2337	84.0
F	169	-1800	109.4
G	-668	-2431	72.5
H	-3560	-4998	28.8
I	-10543	-12554	16.01

$$\% \text{ Difference} \equiv \frac{\sigma(20) - \sigma(12)}{\sigma(20)}$$

TABLE V. FACE NUMBERING CONVENTION FOR PRESSURE LOADS

FACE NUMBER	NATURAL COORDIN.	CORNER NODES N_1 N_2 N_3 N_4	MIDSIDE NODES N_1 N_2 N_3 N_4	α	β
1	(+1, s, t)	1 4 8 5	12 20 16 17	s	t
2	(-1, s, t)	2 3 7 6	10 19 14 18	t	s
3	(r ,+1, t)	1 5 6 2	17 13 18 9	t	r
4	(r ,-1, t)	4 8 7 3	20 15 19 11	r	t
5	(r , s,+1)	1 2 3 4	9 10 11 12	r	s
6	(r , s,-1)	5 6 7 8	13 14 15 16	s	r

APPENDIX A: PREPARATION OF TEMPERATURE TAPE

If temperature input is used with the ADINA program, nodal point temperatures are stored in sequence on device number 56 [4]. At NPS, ITEL 3330 disks have been used, although any storage device may be designated. NSTE + 1 records are read, where NSTE is the number of solution steps. (In linear static analysis, NSTE usually equals 1.)

The read statement in the program is:

```
READ (56) TIME, (TEMPV(I), I=1, NUMNP).
```

On each record, the problem time of the nodal temperatures is read, followed by an implicit "do-loop" from 1 to the total number of node points (NUMNP) to read each nodal point temperature.

A program to establish the temperature tape is listed at the end of this Appendix. It will be discussed in three parts: (A) job control language, (B) the RDADIN subroutine and (C) the binary write.

A. TEMPERATURE TAPE JOB CONTROL LANGUAGE (JCL)

The program listing in Appendix B starts with a two step procedure [17]. The first is a purge operation that returns the previously reserved disk space to the operating system, the second step reserves disk space for the data set which is named F0099.TAPE. One cylinder of 3330 disk space (248K bytes) is reserved on one of the NPS computer center resident

disks. The data set has a variable block size, and an expiration date of 360 days after establishment.

The JCL employed combines ease of continued use of the disk space, with minimum chances of data set expiration.

B. THE RDADIN SUBROUTINE

The RDADIN subroutine was originally programmed as part of the PSAP1 preprocessor. The subroutine will read an ADINA input deck from title card through element connectivity. It generates node numbers, nodal coordinates, element numbers and element connectivities for the entire mesh. Each of the above is stored in a linear array. The programmer is then able to utilize any or all of these arrays to input the nodal point temperature.

For example, in the linear airfoil mesh there were isothermal planes. The matrix of z-coordinates (radial distance from axis of rotation) was used. Each z-coordinate was checked by a series of 'if statements', and nodal temperatures were assigned.

C. THE BINARY WRITE

After the matrix of nodal temperatures was established, the next step was to write it sequentially onto device 56. Array RCD1 contained the problem time for the temperatures, and temperatures for each node; i.e., it was a linear array of dimension NUMNP + 1. This array could then be written onto device 56 by the statement:

```
WRITE (56)RCD1.
```

Device 56 must then be defined in the GO step by:

```
//GO.FT56F001 DD UNIT=3330,VOL=SER-DISK01,
//^DSN=F0099.TAPE,DISP=OLD
```

Each time the WRITE statement is executed one record is established. The sequence must be repeated until NSTE + 1 records are established, where NSTE is the number of solution steps.

In the linear airfoil problem, there was only one solution time step. Therefore, only two records on device 56 were needed. One record was at time=0, and the next record was one time step increment later.

It is recommended that a hard copy of the RCD1 matrix be printed to ascertain the accuracy of the temperature tape.

```
//       (STANDARD GREEN JOB CARD)
//   EXEC   PGM=IEHPROGM
//SYSPRINT   DD   SYSOUT=A
//DISK   DD   UNIT=3330,VOL=SER=DISK01,DISP=SHR
//SYSIN   DD   *
  SCRATCH   VOL=3330=DISK01,DSNAME=F0099.TAPE,PURGE
/*
// EXEC PGM=IEFBR14
//TEMP DD DSN=F0099.TAPE,UNIT=3330,VOL=SER=DISK01,DISP=(NEW,KEEP),
//  LABEL=RETPD=360,SPACE=(CYL,(1,1,1))
/*
// EXEC FORTCLG
//FORT.SYSIN DD *
      IMPLICIT REAL*8(A-H,O-Z)                                          00000010
C*****************************************************************
C
C
C
C       THIS PROGRAM CAN BE USED TO ESTABLISH A TEMPERATURE TAPE
C       FOR THE ADINA CODE. THE RDADIN SUBROUTINE GENERATES THE
C       XPT,YPT,ZPT MATRICES OF COORDINATES. ALSO, THE CONNECTIVITY
C       MATRIX (NCONT),THE ELEMENT NUMBERS AND NODAL NUMBERS
C       ARE STORED IN MATRIX FORM.(NREL AND NUMPT). ANY OF THESE
C       MATRICES CAN BE USED WITH 'IF STATEMENTS' TO ESTABLISH
C       NODAL CONNECTIVITY. THE SUBROUTINE TEMPRT IS PRESENTLY
C       SET UP TO CHECK AGAINST THE Z COCRDINATES.
C
C
C       NODAL TEMPERATURES ARE WRITTEN SEQUENTIALLY CN DEVICE 58
C
C
C*****************************************************************
      COMMON A(15000)                                                   00000020
      COMMON/IPARA/NUMNP,NUMEL                                          00000030
      NUMNP=270
      NUMEL=52
      MTOT=15000                                                        00000060
      N1=1                                                              00000070
      N2=N1+NUMNP+7*NUMEL                                               00000080
      N3=N2+NUMNP+7*NUMEL                                               00000090
      N4=N3+NUMNP+7*NUMEL                                               00000100
      N5=N4+20*NUMEL                                                    00000110
      N6=N5+27*NUMEL                                                    00000120
      N7=N6+NUMNP+7*NUMEL                                               00000130
      N8=N7+NUMEL                                                       00000140
      N9=N8+6*NUMEL                                                     00000150
      N10=N9+6*NUMEL                                                    00000160
```

```
      NLAST=N10+6*NUMEL                                                 00000170
      IF(NLAST.GT.MTOT)GO TO 10                                         00000180
      CALL RDADIN(A(N1),A(N2),A(N3),A(N4),A(N6),A(N7))
      CALL TEMPRT(A(N3),A(N1))
      STOP                                                              00000210
   10 WRITE(6,1000)NLAST                                                00000220
 1000 FORMAT(//5X,'MTOT SHOULD BE INCREASED TO A VALUE OF ',I5//)       00000230
      STOP                                                              00000240
      END                                                               00000250
      SUBROUTINE RDADIN(XPT,YPT,ZPT,NCONT,NREL,NUMPT)                   00001670
C                                                                       00001680
CC    ************************************************************      00001690
C                                                                       00001700
C        SUBROUTINE TO READ ADINA DECK.                                 00001710
C                                                                       00001720
CC    ************************************************************      00001730
C                                                                       00001740
      IMPLICIT REAL*8 (A-H,O-Z)                                         00001750
      REAL*4 TITLE(20)                                                  00001760
      COMMON/IPARA/NUMNP,NUMEL                                          00001770
      COMMON/LOADS/NLOAD,NLCUR,NPTM,IDGRAV                              00001780
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCONT(20,1)       00001790
      DIMENSION IDOF(6),ID(6),IDOLD(6),NCDE(20),NPAR(20)                00001800
     1 ,NP(20),INP(20)                                                  00001810
      DATA CTEST/'X       '/                                            00001820
      NCARD=0                                                           00001830
      READ(5,9000)(TITLE(I),I=1,20)                                     00001840
      WRITE(6,9011)(TITLE(I),I=1,20)                                    00001850
 9011 FORMAT(///5X,20A4,///)                                            00001860
C *** READ MASTER CONTROL CARDS                                         00001870
C *** NUMNP = TOTAL NUMBER OF NODE POINTS                               00001880
C *** NELTYP = NUMBER OF ELEMENT GROUPS                                 00001890
      READ(5,9001) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX,NSTE         00001900
 9001 FORMAT(I5,6I1,I4,3I5)                                             00001910
      NELTYP=NEGL+NEGNL                                                 00001920
      NNODE=NUMNP                                                       00001930
      READ(5,7002) IMASS,IDAMP,IMASSN,IDAMPN                            00001940
 7002 FORMAT(4I5)                                                       00001950
      READ(5,7002) IEIG                                                 00001960
      READ(5,7002) ISREF,NUMREF,IEQUIT,ITEMAX                           00001970
      READ(5,9000) DUMMY                                                00001980
      READ(5,9000) DUMMY                                                00001990
      READ(5,9000) DUMMY                                                00002000
C *** READ OR GENERATE NODAL POINT DATA                                 00002010
      NOLD=0                                                            00002020
      NEQ=0                                                             00002030
   10 READ(5,9006)CT,N,(ID(I),I=1,6),XPT(N),YPT(N),ZPT(N),KN           00002040
 9006 FORMAT(A1,I4,1X,I4,5I5,3F10.0,I5)                                 00002050
```

```
C *** CHECK FOR CYLINDRICAL COORDINATES                                  00002060
      IF(CT.NE.CTEST) GO TO 12                                          00002070
          DUM=ZPT(N)/57.2958                                            00002080
          R=YPT(N)                                                      00002090
          YPT(N)=R*DCOS(ZPT(N)/57.2958D0)                               00002100
          ZPT(N)=R*DSIN(ZPT(N)/57.2958D0)                               00002110
   12 CONTINUE                                                          00002120
      NUMPT(N)=N                                                        00002130
      IF(NOLD.EQ.0) GO TO 50                                            00002140
C *** FOR GENERATION OF FIXED BOUNDARY CONDITIONS                       00002150
      DO 15 I=1,6                                                       00002160
      IF(IDOLD(I).EQ.-1.AND.ID(I).EQ.0) ID(I)=IDOLD(I)                  00002170
   15     CONTINUE                                                      00002180
      IF(KNOLD.EQ.0) GO TO 50                                           00002190
      NUM=(N-NOLD)/KNOLD                                                00002200
      NUMN=NUM-1                                                        00002210
      IF(NUMN.LT.1) GO TO 50                                            00002220
C *** TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS                     00002230
      DO 20 I=1,6                                                       00002240
         IF(IDOF(I).EQ.0.AND.IDOLD(I).EQ.0) NEQ=NEQ+NUMN                00002250
   20 CONTINUE                                                          00002260
      DX=(XPT(N)-XPT(NOLD))/NUM                                         00002270
      IF(CT.NE.CTEST) GO TO 21                                          00002280
      ROLD=YPT(NOLD)/DCOS(DUMOLD)                                       00002290
      RNEW=YPT(N)/DCOS(DUM)                                             00002300
      DR=(RNEW-ROLD)/NUM                                                00002310
      DT=(DUM-DUMOLD)/NUM                                               00002320
      GO TO 22                                                          00002330
   21 CONTINUE                                                          00002340
      DY=(YPT(N)-YPT(NOLD))/NUM                                         00002350
      DZ=(ZPT(N)-ZPT(NOLD))/NUM                                         00002360
   22 CONTINUE                                                          00002370
      K=NOLD                                                            00002380
      DO 30 J=1,NUMN                                                    00002390
         KK=K                                                           00002400
         K=K+KNOLD                                                      00002410
         XPT(K)=XPT(KK)+DX                                              00002420
          IF(CT.NE.CTEST) GO TO 26                                      00002430
          ROLD=ROLD+DR                                                  00002440
          DUMOLD=DUMOLD+DT                                              00002450
          YPT(K)=ROLD*DCOS(DUMOLD)                                      00002460
          ZPT(K)=ROLD*DSIN(DUMOLD)                                      00002470
          GO TO 28                                                      00002480
   26     CONTINUE                                                      00002490
         YPT(K)=YPT(KK)+DY                                              00002500
         ZPT(K)=ZPT(KK)+DZ                                              00002510
   28     CONTINUE                                                      00002520
         NUMPT(K)=K                                                     00002530
```

```
   30     CONTINUE                                                          00002540
   50  NCLD=N                                                               00002550
       KNOLD=KN                                                             00002560
       DUMOLD=DUM                                                           00002570
C ***  TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS                        00002580
       DO 55 I=1,6                                                          00002590
           IF(IDOF(I).EQ.0.AND.ID(I).EQ.0) NEQ=NEQ+1                        00002600
           IDOLD(I)=ID(I)                                                   00002610
   55     CONTINUE                                                          00002620
       IF(N.NE.NUMNP) GO TO 10                                              00002630
C ***  READ LOAD CONTROL CARDS                                              00002640
       READ(5,9012)NLOAD,NLCUR,NPTM,IDGRAV                                  00002650
 9012  FORMAT(4I5)                                                          00002660
       DO 80 I=1,IMASSN                                                     00002670
       IF(IMASSN.EQ.0) GO TO 81                                             00002680
          READ(5,9000) DUMMY                                                00002690
   80     CONTINUE                                                          00002700
   81  CONTINUE                                                             00002710
       IF(IDAMPN.EQ.0) GO TO 91                                             00002720
       DO 90 I=1,IDAMPN                                                     00002730
          READ(5,9000) DUMMY                                                00002740
   90     CONTINUE                                                          00002750
   91  CONTINUE                                                             00002760
C ***  READ INITIAL CONDITIONS                                              00002770
       READ(5,9002) ICON                                                    00002780
       IF(ICON.EQ.0) GO TO 100                                              00002790
       CARDNR=NEQ/6.0                                                       00002800
       NCARD=IDINT(CARDNR)                                                  00002810
       TEST=CARDNR-NCARD                                                    00002820
       IF(TEST.GT.0.1) NCARD=NCARD+1                                        00002830
       DO 95 I=1,NCARD                                                      00002840
          READ(5,9000) DUMMY                                                00002850
   95     CONTINUE                                                          00002860
       IF(IMASS.EQ.0) GO TO 100                                             00002870
       DO 96 I=1,NCARD                                                      00002880
          READ(5,9000) DUMMY                                                00002890
   96     CONTINUE                                                          00002900
       DO 98 I=1,NCARD                                                      00002910
          READ(5,9000) DUMMY                                                00002920
   98     CONTINUE                                                          00002930
 9007  FORMAT(6E12.6)                                                       00002940
  100  CONTINUE                                                             00002950
       NUMEL=0                                                              00002960
       WRITE(6,9009) NEQ,NCARD                                              00002970
 9009  FORMAT(///,' NEQ AND NCARD FOR IC IN GEOM1  = ',I5,10X,I5///)        00002980
C ***  READ ELEMENT CONTROL CARDS                                           00002990
       DO 900 M=1,NELTYP                                                    00003000
           READ(5,9008)(NPAR(I),I=1,20)                                     00003010
```

```
      WRITE(6,9010) (NPAR(I),I=1,20)                                    00003020
 9008 FORMAT(20I4)                                                      00003030
 9010 FORMAT(///,'  NPAR = ',20I5///)                                   00003040
      MTYPE=NPAR(1)                                                     00003050
      NUMMAT=NPAR(16)                                                   00003060
      NSTRES=NPAR(13)                                                   00003070
C *** CALCULATE THE NUMBER OF MATERIAL CASE CARDS                       00003080
      IF(NPAR(15).EQ. 1) NCARD=1                                        00003090
      IF(NPAR(15).EQ. 2) NCARD=2+NPAR(18)                               00003100
      IF(NPAR(15).EQ. 3) NCARD=4                                        00003110
      IF(NPAR(15).EQ. 4) NCARD=4                                        00003120
      IF(NPAR(15).EQ. 5) NCARD=2                                        00003130
      IF(NPAR(15).EQ. 8) NCARD=1                                        00003140
      IF(NPAR(15).EQ. 9) NCARD=1                                        00003150
      IF(NPAR(15).EQ.10) NCARD=6                                        00003160
      IF(NPAR(15).EQ.11) NCARD=6                                        00003170
      IF(NPAR(15).NE.12) GO TO 111                                      00003180
      CARDNR=NPAR(17)/8.0                                               00003190
      NCARD=IDINT(CARDNR)                                               00003200
      TEST=CARDNR-NCARD                                                 00003210
      IF(TEST.GT.0.1) NCARD=NCARD+1                                     00003220
  111 CONTINUE                                                          00003230
      N20=20                                                            00003240
C *** READ MATERIAL PROPERTIES                                          00003250
      DO 222 J=1,NUMMAT                                                 00003260
         READ(5,9000) DUMMY                                             00003270
 9000 FORMAT(20A4)                                                      00003280
         DO 45 I=1,NCARD                                                00003290
            READ(5,9000) DUMMY                                          00003300
   45       CONTINUE                                                    00003310
  222 CONTINUE                                                          00003320
C *** READ STRESS OUTPUT TABLE CARDS                                    00003330
      IF(NPAR(13).EQ.0) GO TO 61                                        00003340
      DO 60 I=1,NSTRES                                                  00003350
         READ(5,9000) DUMMY                                             00003360
   60    CONTINUE                                                       00003370
   61 CONTINUE                                                          00003380
      IF(NPAR(14).EQ.0) NPAR(14)=1                                      00003390
      NEL=NPAR(14)-1                                                    00003400
  130 READ(5,9002) INEL,IINC                                            00003410
 9002 FORMAT(I5,30X,I5)                                                 00003420
      IF(IINC.EQ.0) IINC=1                                              00003430
      READ(5,9004) (INP(I),I=1,8)                                       00003440
      READ(5,9004) (INP(I),I=9,N20)                                     00003450
 9004 FORMAT(12I5)                                                      00003460
  140 NEL=NEL+1                                                         00003470
      NREL(NEL)=NEL                                                     00003480
      ML=INEL-NEL                                                       00003490
```

```
      IF(ML) 150,155,160                                                00003500
  150 WRITE(6,8000)NEL                                                  00003510
 8000 FORMAT(//5X,'WARNING - ERROR IN ELEMENT ',I5//)                   00003520
C *** NC GENERATION OF NODE POINTS REQUIRED                            00003530
  155 DO 156 I=1,N20                                                    00003540
          NP(I)=INP(I)                                                  00003550
          NCONT(I,NEL)=NP(I)                                            00003560
  156     CONTINUE                                                      00003570
      GO TO 162                                                         00003580
C *** GENERATION OF NODE POINTS REQUIRED                               00003590
  160 DO 161 I=1,N20                                                    00003600
          IF(NP(I).EQ.0) GO TO 161                                      00003610
          NP(I)=NP(I)+KN                                                00003620
          NCONT(I,NEL)=NP(I)                                            00003630
  161     CONTINUE                                                      00003640
  162 CONTINUE                                                          00003650
      NUMEL=NUMEL+1                                                     00003660
      IF(NEL.EQ.NPAR(2)) RETURN                                         00003670
      IF(NEL.LT.INEL) GO TO 140                                         00003680
      KN=IINC                                                           00003690
      GO TO 130                                                         00003700
  900     CONTINUE                                                      00003710
      END                                                               00003720
      SUBROUTINE TEMPRT(ZPT,T)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/IPARA/NUMNP,NUMEL
      DIMENSION ZPT(1), T(271)
      EPS=1.D-4
      A=0.
      DO 1000 II=1,2
      DO 100 I=1,NUMNP
      C1=ZPT(I)-3.3850D0
      C1=DABS(C1)
      C2=ZPT(I)-3.5175D0
      C2=DABS(C2)
      C3=ZPT(I)-3.6500D0
      C3=DABS(C3)
      C4=ZPT(I)-3.7825D0
      C4=DABS(C4)
      C5=ZPT(I)-3.9150D0
      C5=DABS(C5)
      C6=ZPT(I)-3.45125D0
      C6=DABS(C6)
      C7=ZPT(I)- 3.58375D0
      C7=DABS(C7)
      C8=ZPT(I)-3.71625D0
      C8=DABS(C8)
      C9=ZPT(I)- 3.84875D0
```

```
      C9=DABS(C9)
      J=I+1
      T(1)=A
      IF(C1.LT.EPS)T(J)=1850.
      IF(C2.LT.EPS)T(J)=1950.
      IF(C3.LT.EPS)T(J)=2000.
      IF(C4.LT.EPS)T(J)=2050.
      IF(C5.LT.EPS)T(J)=2065.
      IF(C6.LT.EPS)T(J)=1900.
      IF(C7.LT.EPS)T(J)=1975.
      IF(C8.LT.EPS)T(J)=2025.
      IF(C9.LT.EPS)T(J)=2060.
100   CONTINUE
      A=A+1.
      WRITE(6,20) T
20    FORMAT(5X,5F20.5)
      WRITE(56) T
1000  CONTINUE
      ENDFILE 56
      RETURN
      END
/*
//GO.FT56F001  DD  UNIT=3330,VOL=SER=DISK01,DSN=F0099.TAPE,DISP=OLD
//GO.SYSIN DD *
        (PUT ADINA DECK HERE)
/*
```

INPUT DECK

```
241   24    4    1    6    0    0    1    1
  7    5  -3.  -3.  -3.  -3.  -3.  -3.  -3.  -3.
  8    5  -3.  -3.  -3.  -3.  -3.  -3.  -3.  -3.
  9    5  -3.  -3.  -3.  -3.  -3.  -3.  -3.  -3.
 10    5  -3.  -3.  -3.  -3.  -3.  -3.  -3.  -3.
 11    5  -3.  -3.  -3.  -3.  -3.  -3.  -3.  -3.
 12    5  -3.  -3.  -3.  -3.  -3.  -3.  -3.  -3.
PRESSURE LOAD VERIFICATION - BENDING SPECIMEN CASE 4
2410OC111    1    0    1    1        .1E1          0.    0    0    0          1    1
   0    0    0    0
   0
   0    0    0    0         0
        0         0         0
   0

   9    0    0    0    1    1    1          0      .1250      .1250   19
 237    1    0    0    1    1    1        1.5      .1250      .1250
   1    0    0    0    1    1    1          0      .0625      .1250   19
 229    1    0    0    1    1    1        1.5      .0625      .1250
   4    0    1    0    1    1    1          0          0      .1250   19
 232    1    1    0    1    1    1        1.5          0      .1250
  10    0    0    0    1    1    1          0      .1250     .09375   19
 238    1    0    0    1    1    1        1.5      .1250     .09375
   5    0    1    0    1    1    1          0          0     .09375   19
 233    1    1    0    1    1    1        1.5          0     .09375
  11    0    0    0    1    1    1          0      .1250      .0625   19
 239    1    0    0    1    1    1        1.5      .1250      .0625
   2    0    0    0    1    1    1          0      .0625      .0625   19
 230    1    0    0    1    1    1        1.5      .0625      .0625
   6    0    1    0    1    1    1          0          0      .0625   19
 234    1    1    0    1    1    1        1.5          0      .0625
  12    0    0    0    1    1    1          0      .1250     .03125   19
 240    1    0    0    1    1    1        1.5      .1250     .03125
   7    0    1    0    1    1    1          0          0     .03125   19
 235    1    1    0    1    1    1        1.5          0     .03125
   3    0    0    0    1    1    1          0      .0625          0   19
 117    0    0    1    1    1    1       .750      .0625          0
 136    0    0    0    1    1    1       .875      .0625          0   19
 231    1    0    0    1    1    1        1.5      .0625          0
   8    0    1    0    1    1    1          0          0          0   19
 122    0    1    1    1    1    1       .750          0          0
 141    0    1    0    1    1    1       .875          0          0   19
 236    1    1    0    1    1    1        1.5          0          0
  17    0    0    0    1    1    1      .0625      .1250      .1250   19
 226    0    0    0    1    1    1    1.4375      .1250      .1250
  14    0    0    0    1    1    1      .0625          0      .1250   19
```

```
223     0     0     0     1     1     1     1.4375          0      .1250
 18     0     0     0     1     1     1      .0625       .1250      .0625    19
227     0     0     0     1     1     1     1.4375       .1250      .0625
 15     0     0     0     1     1     1      .0625          0       .0625    19
224     0     0     0     1     1     1     1.4375          0       .0625
 19     0     0     0     1     1     1      .0625       .1250          0    19
228     0     0     0     1     1     1     1.4375       .1250          0
 16     0     0     0     1     1     1      .0625          0           0    19
225     0     0     0     1     1     1     1.4375          0           0
 13     0     0     0     1     1     1           0      .1250          0    19
127     0     0     1     1     1     1        .750      .1250          0
146     0     0     0     1     1     1        .875      .1250          0    19
241     1     0     0     1     1     1         1.5      .1250          0
  6     1     2     0
  0     1
  3    24     0     0          20           4     4          1          1     1     0     0
  1  .2992D-3
   45.E6      .2000
  1     2     3     4     5     6     7     8
  1     0     0     1     1     0     0    19
 28     9     4    23    30    11     6    25
 17     1    14    20    18     2    15    21    29    10     5    24
 12     0     0     1     1     0     0
237   218   213   232   239   220   215   234
226   210   223   229   227   211   224   230   238   219   214   233
 13     0     0     1     1     0     0    19
 30    11     6    25    32    13     8    27
 18     2    15    21    19     3    16    22    31    12     7    26
 24     0     0     1     1     0     0
239   220   215   234   241   222   217   236
227   211   224   230   228   212   225   231   240   221   216   235
```

ICHK1= 0
ICHK2= 1
ICHK3= 1
ICHK4= 1

TOTAL NUMBER OF NODE POINTS = 241
TOTAL NUMBER OF ELEMENTS = 24
THE NUMBER OF GAUSS POINTS = 4
THE NUMBER OF PRESSURE LOADS APPLIED = 6

THE VALUE OF MTOT IS 2258

PRESSURE LOAD VERIFICATION - BENDING SPECIMEN CASE 4

NEQ AND NCARD FOR IC IN GEOM1 = 642 0

NPAR = 3 24 0 0 0 0 20 0 0 4 4 0 1 0

NODE	X COMPONENT	Y COMPONENT	Z COMPONENT
115	0.1781559551300-17	-0.1977433770090-18	-0.1562500000000-01
118	0.3559979448460-18	0.4929568945070-18	0.3906250000000-02
123	0.2396087128870-18	-0.1978386865310-18	0.3906250000000-02
128	0.3278715892400-18	0.1241846955770-17	-0.1562500000000-01
131	0.1614959145910-18	-0.1883770961820-17	-0.1562500000000-01
134	0.4893675038300-18	-0.1011425686710-17	-0.3125000000000-01
137	-0.2018330315350-18	0.1137590555650-17	0.7812500000000-02
142	-0.2235772684450-18	-0.6278534200360-18	0.7812500000000-02
147	0.3278715892400-18	0.1241846955770-17	-0.1562500000000-01
150	0.1614959145910-18	-0.1883770961820-17	-0.1562500000000-01
153	0.4893675038300-18	-0.1011425686710-17	-0.3125000000000-01
156	-0.2018330315350-18	0.1137590555650-17	0.7812500000000-02
161	-0.2235772684450-18	-0.6278534200360-18	0.7812500000000-02
166	0.3278715892400-18	0.1241846955770-17	-0.1562500000000-01
169	0.1614959145910-18	-0.1883770961820-17	-0.1562500000000-01
172	0.4893675038300-18	-0.1011425686710-17	-0.3125000000000-01
175	-0.2018330315350-18	0.1137590555650-17	0.7812500000000-02
180	-0.2235772684450-18	-0.6278534200360-18	0.7812500000000-02
185	0.3278715892400-18	0.1241846955770-17	-0.1562500000000-01
188	0.1614959145910-18	-0.1883770961820-17	-0.1562500000000-01
191	0.4893675038300-18	-0.1011425686710-17	-0.3125000000000-01
194	-0.2018330315350-18	0.1137590555650-17	0.7812500000000-02
199	-0.2235772684450-18	-0.6278534200360-18	0.7812500000000-02
204	0.3278715892400-18	0.1241846955770-17	-0.1562500000000-01
207	0.1614959145910-18	-0.1883770961820-17	-0.1562500000000-01
210	0.4893675038300-18	-0.1011425686710-17	-0.3125000000000-01
213	-0.2018330315350-18	0.1137590555650-17	0.7812500000000-02
218	-0.2235772684450-18	-0.6278534200360-18	0.7812500000000-02
223	0.3278715892400-18	0.1241846955770-17	-0.1562500000000-01
226	0.1614959145910-18	-0.1883770961820-17	-0.1562500000000-01
229	-0.1292192047470-17	-0.8136823096990-18	-0.1562500000000-01
232	-0.5578309763800-18	0.6446336611460-18	0.3906250000000-02
237	-0.4631859813310-18	-0.4300147335050-18	0.3906250000000-02

NODE	DOF	NCUR	FACTOR
115	3	1	-0.1562500C000000006D-01
118	3	1	0.3906250000000048D-02
123	3	1	0.3906250000000044D-02
128	3	1	-0.1562500000000004D-01
131	3	1	-0.1562500000000004D-01
134	3	1	-0.3125000000000008D-01
137	3	1	0.7812500000000098D-02
142	3	1	0.7812500000000103D-02
147	3	1	-0.1562500000000004D-01
150	3	1	-0.1562500000000003D-01
153	3	1	-0.3125000000000006D-01
156	3	1	0.7812500000000103D-02
161	3	1	0.7812500000000105D-02
166	3	1	-0.1562500C00000004D-01
169	3	1	-0.1562500000000003D-01
172	3	1	-0.3125000000000010D-01
175	3	1	0.7812500000000092D-02
180	3	1	0.7812500000000098D-02
185	3	1	-0.1562500C00000005D-01
188	3	1	-0.1562500000000004D-01
191	3	1	-0.3125000C00000009D-01
194	3	1	0.7812500000000103D-02
199	3	1	0.7812500000000109D-02
204	3	1	-0.1562500000000005D-01
207	3	1	-0.1562500C00000004D-01
210	3	1	-0.3125000000000007D-01
213	3	1	0.7812500C00000096D-02
218	3	1	0.7812500000000106D-02
223	3	1	-0.1562500000000003D-01
226	3	1	-0.1562500C00000002D-01
229	3	1	-0.1562499999999999D-01
232	3	1	0.3906250000000058D-02
237	3	1	0.3906250000000057D-02

THE TOTAL NUMBER OF LOADS IS 33

```
      IMPLICIT REAL*8(A-H,O-Z)                                          PRE00010
C*****************************************************************     PRE00020
C                                                                       PRE00030
C                                                                       PRE00040
C     THIS PROGRAM IS A FINITE ELEMENT PREPROCESSOR THAT CAN            PRE00050
C     BE USED TO CALCULATE CONSISTENT PRESSURE LOADS.                   PRE00060
C                                                                       PRE00070
C                                                                       PRE00080
C     1. CONTROL CARD  (10I5)                                           PRE00090
C                                                                       PRE00100
C     COLUMNS          VARIABLE          MEANING                        PRE00110
C                                                                       PRE00120
C      1-5             NUMNP             TOTAL NUMBER OF NODES          PRE00130
C                                                                       PRE00140
C      6-10            NUMEL             TOTAL NUMBER OF ELEMENTS       PRE00150
C                                                                       PRE00160
C      11-15           NGP               NUMBER OF GAUSS POINTS         PRE00170
C                                        USED IN INTEGRATION.(2-6)      PRE00180
C                                                                       PRE00190
C      16-20           NCUR              NUMBER OF ADINA LOAD CURVE     PRE00200
C                                        THAT OUTPUT LOADS REFER TO.    PRE00210
C                                        (USUALLY NCUR=1)               PRE00220
C                                                                       PRE00230
C      21-25           NPLOAD            NUMBER OF ELEMENT FACES WITH   PRE00240
C                                        DISTRIBUTED LOADS.             PRE00250
C                                                                       PRE00260
C      26-30           IPRINT1           PRINT CONTROL                  PRE00270
C                                                                       PRE00280
C      31-35           IPRINT2           PRINT CONTROL                  PRE00290
C                                                                       PRE00300
C      36-40           IPRINT3           PRINT CONTROL                  PRE00310
C                                                                       PRE00320
C      41-45           IPRINT4           PRINT CONTROL                  PRE00330
C                                                                       PRE00340
C                                                                       PRE00350
C                                                                       PRE00360
C     2. NODAL PRESSURES CARD (2I5,8F10.5)                              PRE00370
C        INPUT NPLOAD OF THESE CARDS                                    PRE00380
C                                                                       PRE00390
C     COLUMNS          VARIABLE          MEANING                        PRE00400
C                                                                       PRE00410
C      1-5             NPREL(J)          ELEMENT NUMBER                 PRE00420
C                                                                       PRE00430
C      6-10            NFACE(J)          FACE TO WHICH PRESSURE         PRE00440
C                                        APPLIED                        PRE00450
C                                                                       PRE00460
C      11-50           PRESS(I,J)        NODAL PRESSURE INTENSITIES     PRE00470
C                                        INPUT HERE. CORNER NODES       PRE00480
```

```
C                                                  FIRST,THEN MIDSIDE NODES.      PRE00490
CC                                                 POSITIVE PRESSURE WILL TEND    PRE00500
CCC                                                TO COMPRESS ELEMENT.           PRE00510
CCCC                                                                              PRE00520
CCCCC                                                                             PRE00530
CCCCC      PRINT CONTROL                                                          PRE00540
CCCCC                                                                             PRE00550
CCCCC   ICHK1: IF(ICHK1.EQ.1) PROGRAM WILL JUST READ AND PRINT ADINA              PRE00560
CCCCC          DECK. WILL NOT CALL SUBROUTINE TRANS.                              PRE00570
CCCCC                                                                             PRE00580
C       ICHK2: IF(ICHK2.EQ.1) PROGRAM WILL NOT PRINT ELEMENT-BY-                  PRE00590
CC             ELEMENT RESULTS.                                                   PRE00600
CCC                                                                               PRE00610
CCCC    ICHK3: IF (ICHK3.EQ.1)PROGRAM WILL NOT PUNCH DECK                         PRE00620
CCCCC                                                                             PRE00630
CCCCC   ICHK4: IF(ICHK4.EQ.1)PROGRAM WILL SKIP PRINTING INPUT INFORM-             PRE00640
CCCCC          ATION. (WILL NOT CALL SUBROUTINE PNTOUT)                           PRE00650
CCCCC                                                                             PRE00660
CCCCC     JOHN PREISEL     NOVEMBER 1977    NAVAL POSTGRADUATE SCHOOL            PRE00670
CCCCC                      MONTEREY,CALIFORNIA 93940                            PRE00680
CCCCC                                                                             PRE00690
CCCCC***************************************************************************PRE00700
CCCCC                                                                             PRE00710
CCCC                                                                              PRE00720
CCC                                                                               PRE00730
      COMMON A(15000)                                                             PRE00740
      COMMON/IPARA/NUMNP,NUMEL,NGP,NCUR,NPLOAD                                    PRE00750
      COMMON/CHECK/ICHK1,ICHK2,ICHK3,ICHK4                                        PRE00760
      MTOT=15000                                                                  PRE00770
      READ(5,1001)NUMNP,NUMEL,NGP,NCUR,NPLOAD,ICHK1,ICHK2,ICHK3,                 PRE00780
     1ICHK4                                                                       PRE00790
      WRITE(6,1002)ICHK1,ICHK2,ICHK3,ICHK4                                        PRE00800
1002  FORMAT(///5X,'ICHK1=',I5//5X,'ICHK2=',I5//5X,'ICHK3=',I5//                  PRE00810
     15X,'ICHK4=',I5///)                                                          PRE00820
1001  FORMAT(10I5)                                                                PRE00830
      WRITE(6,1003)NUMNP,NUMEL,NGP,NPLOAD                                         PRE00840
1003  FORMAT(//5X,'TOTAL NUMBER OF NODE POINTS =',I5//5X,'TOTAL NUMBER            PRE00850
     1OF ELEMENTS = ',I5//5X,'THE NUMBER OF GAUSS POINTS =',I5//5X,'THE          PRE00860
     2NUMBER OF PRESSURE LOADS APPLIED = ',I5///)                                 PRE00870
      N1=1                                                                        PRE00880
      N2=N1+3*NUMNP                                                               PRE00890
      N3=N2+NUMNP                                                                 PRE00900
      N4=N3+NUMNP                                                                 PRE00910
      N5=N4+NUMNP                                                                 PRE00920
      N6=N5+NUMNP                                                                 PRE00930
      N7=N6+NUMEL                                                                 PRE00940
      N8=N7+20*NUMEL                                                              PRE00950
      N9=N8+8*NPLOAD                                                              PRE00960
```

```
      N10=N9+NPLOAD                                                     PRE00970
      NLAST=N10+2*NPLOAD                                                PRE00980
      IF(NLAST.LT.MTOT)WRITE(6,2001)NLAST                               PRE00990
2001  FORMAT(//5X,'THE VALUE OF MTOT IS  ',I7///)                       PRE01000
      IF(NLAST.GT.MTOT)WRITE(6,2000)NLAST                               PRE01010
2000  FORMAT(//5X,'MTOT SHOULD BE INCREASED TO  ',I7//)                 PRE01020
      CALL CENTF(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9), PRE01030
     1A(N10))                                                           PRE01040
      STOP                                                              PRE01050
      END                                                               PRE01060
      SUBROUTINE CENTF(TFVT,XPT,YPT,ZPT,NUMPT,NREL,NCONT,PRESS,NPREL,   PRE01070
     1NNFACE)                                                           PRE01080
C                                                                       PRE01090
CC                                                                      PRE01100
C                                                                       PRE01110
C***********************************************************************PRE01120
C                                                                       PRE01130
C     SUBROUTINE TO COORDINATE I/O FOR CONSISTENT PRESSURE LOADING      PRE01140
C                                                                       PRE01150
C***********************************************************************PRE01160
CC                                                                      PRE01170
C                                                                       PRE01180
      IMPLICIT REAL*8 (A-H,O-Z)                                         PRE01190
      COMMON/CHECK/ICHK1,ICHK2,ICHK3,ICHK4                              PRE01200
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1)                   PRE01210
      DIMENSION TFVT(3,1),PRESS(8,1),NCONT(20,1),NNFACE(1),NPREL(1)     PRE01220
      CALL RDADIN(XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)      PRE01230
40    IF (ICHK4.EQ.1)GO TO 10                                           PRE01240
      CALL PNTOUT(XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)      PRE01250
      IF (ICHK1.EQ.1)RETURN                                             PRE01260
10    CONTINUE                                                          PRE01270
      CALL TRANS(TFVT,XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)  PRE01280
      RETURN                                                            PRE01290
      END                                                               PRE01300
      SUBROUTINE PNTOUT(XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)PRE01310
C                                                                       PRE01320
C***********************************************************************PRE01330
C                                                                       PRE01340
C     SUBROUTINE TO PRINT OUT NODE AND ELEMENT INFORMATION              PRE01350
C                                                                       PRE01360
C***********************************************************************PRE01370
CC                                                                      PRE01380
C                                                                       PRE01390
      IMPLICIT REAL*8(A-H,O-Z)                                          PRE01400
      COMMON/IPARA/NUMNP,NUMEL,NGP,NCUR,NPLOAD                          PRE01410
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCONT(20,1)       PRE01420
      DIMENSION PRESS(8,1),NPREL(1),NNFACE(1)                           PRE01430
C                                                                       PRE01440
```

```
C *** FOR OUTPUT OF GEOMETRY INFORMATION                                  PRE01450
C                                                                         PRE01460
      WRITE(6,16)                                                         PRE01470
   16 FORMAT(///,5X,'GRID POINT INFORMATION',///)                         PRE01480
      WRITE(6,17)                                                         PRE01490
   17 FORMAT(5X,'RESEQUENCED',4X,'USER INPUT'/                            PRE01500
     15X,'GRID POINT',5X,'GRID POINT'/                                    PRE01510
     25X,'NUMBER',9X,'NUMBER',13X,'X',14X,'Y',14X,'Z'//)                  PRE01520
      DO 30 I=1,NUMNP                                                     PRE01530
      WRITE(6,18) I,NUMPT(I),XPT(I),YPT(I),ZPT(I)                         PRE01540
   18 FORMAT(2X,I10,5X,I10,3X,3E15.4)                                     PRE01550
   30 CONTINUE                                                            PRE01560
      WRITE(6,19)                                                         PRE01570
   19 FORMAT(///,5X,'ELEMENT INFORMATION - WITH RESEQUENCED GRID POINTS   PRE01580
     1'///)                                                               PRE01590
      WRITE (6,9008)                                                      PRE01600
 9008 FORMAT(1X,'RESEQUENCED',4X,'USER INPUT',25X,'GRID POINTS'/          PRE01610
     11X,'ELEMENT',8X,'ELEMENT'/                                          PRE01620
     21X,'NUMBER',9X,'NUMBER',7X,'    1    2    3    4    5    6    7   PRE01630
     3 8    9   10   11   12   13   14   15   16   17   18   19   20'//)PRE01640
      DO 40 I=1,NUMEL                                                     PRE01650
      WRITE(6,9010)I,NREL(I),(NCONT(J,I),J=1,20)                          PRE01660
 9010 FORMAT(1X,I4,11X,I4,9X,20I5)                                        PRE01670
40    CONTINUE                                                            PRE01680
      WRITE(6,20)                                                         PRE01690
20    FORMAT(///5X,'PRESSURE LOADING INFORMATION   '///)                  PRE01700
      WRITE(6,21)                                                         PRE01710
21    FORMAT(5X,'EL. NO.',5X,'NFACE',5X,'NGP',20X,'PRESSURE AT NODES'//)PRE01720
      DO 23 J=1,NPLOAD                                                    PRE01730
      WRITE(6,22)NPREL(J),NNFACE(J),NGP,(PRESS(I,J),I=1,8)                PRE01740
22    FORMAT(2(5X,I5),3X,I5,3X,8F10.5)                                    PRE01750
23    CONTINUE                                                            PRE01760
999   RETURN                                                              PRE01770
      END                                                                 PRE01780
      SUBROUTINE RDADIN(XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFACE)PRE01790
C                                                                         PRE01800
C     ***************************************************************   PRE01810
C                                                                         PRE01820
C        SUBROUTINE TO READ ADINA DECK.                                   PRE01830
C                                                                         PRE01840
C     ***************************************************************   PRE01850
C                                                                         PRE01860
      IMPLICIT REAL*8 (A-H,O-Z)                                           PRE01870
      REAL*4 TITLE(20)                                                    PRE01880
      COMMON/IPARA/NUMNP,NUMEL,NGP,NCUR,NPLOAD                            PRE01890
      COMMON/LOADS/NLOAD,NLCUR,NPTM,IDGRAV                                PRE01900
      DIMENSION XPT(1),YPT(1),ZPT(1),NUMPT(1),NREL(1),NCONT(20,1)         PRE01910
      DIMENSION PRESS(8,1),NPREL(1),NNFACE(1)                             PRE01920
```

```
      DIMENSION IDOF(6),ID(6),IDOLD(6),NODE(20),NPAR(20)                PRE01930
     1 ,NP(20),INP(20)                                                  PRE01940
      DATA CTEST/'X       '/                                            PRE01950
      NCARD=0                                                           PRE01960
C                                                                       PRE01970
C*****READ PRESSURE LOAD DATA********                                   PRE01980
C                                                                       PRE01990
      DO 9015 J=1,NPLOAD                                                PRE02000
9014  FORMAT(2I5,8F5.0)                                                 PRE02010
9015  READ(5,9014)NPREL(J),NNFACE(J),(PRESS(I,J),I=1,8)                 PRE02020
      READ(5,9000)(TITLE(I),I=1,20)                                     PRE02030
      WRITE(6,9011)(TITLE(I),I=1,20)                                    PRE02040
 9011 FORMAT(///5X,20A4,///)                                            PRE02050
C *** READ MASTER CONTROL CARDS                                         PRE02060
C *** NUMNP = TOTAL NUMBER OF NODE POINTS                               PRE02070
C *** NELTYP = NUMBER OF ELEMENT GROUPS                                 PRE02080
      READ(5,9001) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX,NSTE         PRE02090
 9001 FORMAT(I5,6I1,I4,3I5)                                             PRE02100
      NELTYP=NEGL+NEGNL                                                 PRE02110
      NNODE=NUMNP                                                       PRE02120
      READ(5,7002) IMASS,IDAMP,IMASSN,IDAMPN                            PRE02130
 7002 FORMAT(4I5)                                                       PRE02140
      READ(5,7002) IEIG                                                 PRE02150
      READ(5,7002) ISREF,NUMREF,IEQUIT,ITEMAX                           PRE02160
      READ(5,9000) DUMMY                                                PRE02170
      READ(5,9000) DUMMY                                                PRE02180
      READ(5,9000) DUMMY                                                PRE02190
C *** READ OR GENERATE NODAL POINT DATA                                 PRE02200
      NOLD=0                                                            PRE02210
      NEG=0                                                             PRE02220
   10 READ(5,9006)CT,N,(ID(I),I=1,6),XPT(N),YPT(N),ZPT(N),KN            PRE02230
 9006 FORMAT(A1,I4,1X,I4,5I5,3F10.0,I5)                                 PRE02240
C *** CHECK FOR CYLINDRICAL COORDINATES                                 PRE02250
      IF(CT.NE.CTEST) GO TO 12                                          PRE02260
          DUM=ZPT(N)/57.2958                                            PRE02270
          R=YPT(N)                                                      PRE02280
          YPT(N)=R*DCOS(ZPT(N)/57.2958D0)                               PRE02290
          ZPT(N)=R*DSIN(ZPT(N)/57.2958D0)                               PRE02300
   12 CONTINUE                                                          PRE02310
      NUMPT(N)=N                                                        PRE02320
      IF(NOLD.EQ.0) GO TO 50                                            PRE02330
C *** FOR GENERATION OF FIXED BOUNDARY CONDITIONS                       PRE02340
      DO 15 I=1,6                                                       PRE02350
      IF(IDOLD(I).EQ.-1.AND.ID(I).EQ.0) ID(I)=IDOLD(I)                  PRE02360
   15     CONTINUE                                                      PRE02370
      IF(KNOLD.EQ.0) GO TO 50                                           PRE02380
      NUM=(N-NOLD)/KNOLD                                                PRE02390
      NUMN=NUM-1                                                        PRE02400
```

```
      IF(NUMN.LT.1) GO TO 50                                            PRE02410
C *** TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS                     PRE02420
      DO 20 I=1,6                                                       PRE02430
         IF(IDOF(I).EQ.0.AND.IDOLD(I).EQ.0) NEQ=NEQ+NUMN                PRE02440
   20 CONTINUE                                                          PRE02450
      DX=(XPT(N)-XPT(NOLD))/NUM                                         PRE02460
      IF(CT.NE.CTEST) GO TO 21                                          PRE02470
      ROLD=YPT(NOLD)/DCOS(DUMOLD)                                       PRE02480
      RNEW=YPT(N)/DCOS(DUM)                                             PRE02490
      DR=(RNEW-ROLD)/NUM                                                PRE02500
      DT=(DUM-DUMOLD)/NUM                                               PRE02510
      GO TO 22                                                          PRE02520
   21 CONTINUE                                                          PRE02530
      DY=(YPT(N)-YPT(NOLD))/NUM                                         PRE02540
      DZ=(ZPT(N)-ZPT(NOLD))/NUM                                         PRE02550
   22 CONTINUE                                                          PRE02560
      K=NOLD                                                            PRE02570
      DO 30 J=1,NUMN                                                    PRE02580
         KK=K                                                           PRE02590
         K=K+KNOLD                                                      PRE02600
         XPT(K)=XPT(KK)+DX                                              PRE02610
          IF(CT.NE.CTEST) GO TO 26                                      PRE02620
          ROLD=ROLD+DR                                                  PRE02630
          DUMOLD=DUMOLD+DT                                              PRE02640
          YPT(K)=ROLD*DCOS(DUMOLD)                                      PRE02650
          ZPT(K)=ROLD*DSIN(DUMOLD)                                      PRE02660
          GO TO 28                                                      PRE02670
   26     CONTINUE                                                      PRE02680
         YPT(K)=YPT(KK)+DY                                              PRE02690
         ZPT(K)=ZPT(KK)+DZ                                              PRE02700
   28     CONTINUE                                                      PRE02710
         NUMPT(K)=K                                                     PRE02720
   30    CONTINUE                                                      PRE02730
   50 NOLD=N                                                            PRE02740
      KNOLD=KN                                                          PRE02750
      DUMOLD=DUM                                                        PRE02760
C *** TO COUNT DOFS TO DETERMINE NUMBER OF IC CARDS                     PRE02770
      DO 55 I=1,6                                                       PRE02780
          IF(IDOF(I).EQ.0.AND.ID(I).EQ.0) NEQ=NEQ+1                     PRE02790
          IDOLD(I)=ID(I)                                                PRE02800
   55    CONTINUE                                                      PRE02810
      IF(N.NE.NUMNP) GO TO 10                                           PRE02820
C *** READ LOAD CONTROL CARDS                                           PRE02830
      READ(5,9012)NLOAD,NLCUR,NPTM,IDGRAV                               PRE02840
 9012 FORMAT(4I5)                                                       PRE02850
      DO 80 I=1,IMASSN                                                  PRE02860
      IF(IMASSN.EQ.0) GO TO 81                                          PRE02870
         READ(5,9000) DUMMY                                             PRE02880
```

```
   80     CONTINUE                                                          PRE02890
   81  CONTINUE                                                             PRE02900
       IF(IDAMPN.EQ.0) GO TO 91                                             PRE02910
       DO 90 I=1,IDAMPN                                                     PRE02920
          READ(5,9000) DUMMY                                                PRE02930
   90     CONTINUE                                                          PRE02940
   91  CONTINUE                                                             PRE02950
C  *** READ INITIAL CONDITIONS                                              PRE02960
       READ(5,9002) ICON                                                    PRE02970
       IF(ICON.EQ.0) GO TO 100                                              PRE02980
       CARDNR=NEQ/6.0                                                       PRE02990
       NCARD=IDINT(CARDNR)                                                  PRE03000
       TEST=CARDNR-NCARD                                                    PRE03010
       IF(TEST.GT.0.1) NCARD=NCARD+1                                        PRE03020
       DO 95 I=1,NCARD                                                      PRE03030
          READ(5,9000) DUMMY                                                PRE03040
   95     CONTINUE                                                          PRE03050
       IF(IMASS.EQ.0) GO TO 100                                             PRE03060
       DO 96 I=1,NCARD                                                      PRE03070
          READ(5,9000) DUMMY                                                PRE03080
   96     CONTINUE                                                          PRE03090
       DO 98 I=1,NCARD                                                      PRE03100
          READ(5,9000) DUMMY                                                PRE03110
   98     CONTINUE                                                          PRE03120
 9007  FORMAT(6E12.6)                                                       PRE03130
  100  CONTINUE                                                             PRE03140
       NUMEL=0                                                              PRE03150
       WRITE(6,9009) NEQ,NCARD                                              PRE03160
 9009  FORMAT(///,' NEQ AND NCARD FOR IC IN GEOM1  = ',I5,10X,I5///)        PRE03170
C  *** READ ELEMENT CONTROL CARDS                                           PRE03180
       DO 900 M=1,NELTYP                                                    PRE03190
          READ(5,9008)(NPAR(I),I=1,20)                                      PRE03200
       WRITE(6,9010) (NPAR(I),I=1,20)                                       PRE03210
 9008  FORMAT(20I4)                                                         PRE03220
 9010  FORMAT(///,'  NPAR = ',20I5///)                                      PRE03230
          MTYPE=NPAR(1)                                                     PRE03240
       NUMMAT=NPAR(16)                                                      PRE03250
       NSTRES=NPAR(13)                                                      PRE03260
C  *** CALCULATE THE NUMBER OF MATERIAL CASE CARDS                          PRE03270
       IF(NPAR(15).EQ. 1) NCARD=1                                           PRE03280
       IF(NPAR(15).EQ. 2) NCARD=2+NPAR(18)                                  PRE03290
       IF(NPAR(15).EQ. 3) NCARD=4                                           PRE03300
       IF(NPAR(15).EQ. 4) NCARD=4                                           PRE03310
       IF(NPAR(15).EQ. 5) NCARD=2                                           PRE03320
       IF(NPAR(15).EQ. 8) NCARD=1                                           PRE03330
       IF(NPAR(15).EQ. 9) NCARD=1                                           PRE03340
       IF(NPAR(15).EQ.10) NCARD=6                                           PRE03350
       IF(NPAR(15).EQ.11) NCARD=6                                           PRE03360
```

```
      IF(NPAR(15).NE.12) GO TO 111                                      PRE03370
      CARDNR=NPAR(17)/8.0                                               PRE03380
      NCARD=IDINT(CARDNR)                                               PRE03390
      TEST=CARDNR-NCARD                                                 PRE03400
      IF(TEST.GT.0.1) NCARD=NCARD+1                                     PRE03410
  111 CONTINUE                                                          PRE03420
      N20=20                                                            PRE03430
C *** READ MATERIAL PROPERTIES                                          PRE03440
      DO 222 J=1,NUMMAT                                                 PRE03450
         READ(5,9000) DUMMY                                             PRE03460
 9000 FORMAT(20A4)                                                      PRE03470
         DO 45 I=1,NCARD                                                PRE03480
            READ(5,9000) DUMMY                                          PRE03490
   45       CONTINUE                                                    PRE03500
  222 CONTINUE                                                          PRE03510
C *** READ STRESS OUTPUT TABLE CARDS                                    PRE03520
      IF(NPAR(13).EQ.0) GO TO 61                                        PRE03530
      DO 60 I=1,NSTRES                                                  PRE03540
         READ(5,9000) DUMMY                                             PRE03550
   60    CONTINUE                                                       PRE03560
   61 CONTINUE                                                          PRE03570
      IF(NPAR(14).EQ.0) NPAR(14)=1                                      PRE03580
      NEL=NPAR(14)-1                                                    PRE03590
  130 READ(5,9002) INEL,IINC                                            PRE03600
 9002 FORMAT(I5,30X,I5)                                                 PRE03610
      IF(IINC.EQ.0) IINC=1                                              PRE03620
      READ(5,9004) (INP(I),I=1,8)                                       PRE03630
      READ(5,9004) (INP(I),I=9,N20)                                     PRE03640
 9004 FORMAT(12I5)                                                      PRE03650
  140 NEL=NEL+1                                                         PRE03660
      NREL(NEL)=NEL                                                     PRE03670
      ML=INEL-NEL                                                       PRE03680
      IF(ML) 150,155,160                                                PRE03690
  150 WRITE(6,8000)NEL                                                  PRE03700
 8000 FORMAT(//5X,'WARNING - ERROR IN ELEMENT ',I5//)                   PRE03710
C *** NO GENERATION OF NODE POINTS REQUIRED                             PRE03720
  155 DO 156 I=1,N20                                                    PRE03730
         NP(I)=INP(I)                                                   PRE03740
         NCONT(I,NEL)=NP(I)                                             PRE03750
  156    CONTINUE                                                       PRE03760
      GO TO 162                                                         PRE03770
C *** GENERATION OF NODE POINTS REQUIRED                                PRE03780
  160 DO 161 I=1,N20                                                    PRE03790
         IF(NP(I).EQ.0) GO TO 161                                       PRE03800
         NP(I)=NP(I)+KN                                                 PRE03810
         NCONT(I,NEL)=NP(I)                                             PRE03820
  161    CONTINUE                                                       PRE03830
  162 CONTINUE                                                          PRE03840
```

```
      NUMEL=NUMEL+1                                                     PRE03850
      IF(NEL.EQ.NPAR(2)) RETURN                                         PRE03860
      IF(NEL.LT.INEL) GO TO 140                                         PRE03870
      KN=IINC                                                           PRE03880
      GO TO 130                                                         PRE03890
  900     CONTINUE                                                      PRE03900
      END                                                               PRE03910
      SUBROUTINE TRANS(TFVT,XPT,YPT,ZPT,NCONT,NREL,NUMPT,PRESS,NPREL,NNFPRE03920
     1ACE)                                                              PRE03930
C                                                                       PRE03940
C                                                                       PRE03950
C*******************************************************************PRE03960
C                                                                       PRE03970
C   MAIN PROGRAM USED IN CALCULATION OF CONSISTENT FORCE LOADING FOR    PRE03980
C   DISTRIBUTED FORCE LOADS                                             PRE03990
C                                                                       PRE04000
C*******************************************************************PRE04010
      IMPLICIT REAL*8(A-H,O-Z)                                          PRE04020
      DIMENSION COD(20,3),NCON(20),PWA(8),TITL1(10),ACOD(20,3)          PRE04030
      DIMENSION TFVT(3,1),XPT(1),YPT(1),ZPT(1),NCONT(20,1),PRESS(8,1)   PRE04040
      DIMENSION NPREL(1),FVT(60),NODES(8),NNFACE(1)                     PRE04050
      DIMENSION TIT(10)                                                 PRE04060
      COMMON/CHECK/ICHK1,ICHK2,ICHK3,ICHK4                              PRE04070
      COMMON/GEOM/COD,NCON,IEL                                          PRE04080
      COMMON /IPARA/NUMNP,NUMEL,NGP,NCUR,NPLOAD                         PRE04090
      COMMON/PRESR/PWA,WFAC,NFACE                                       PRE04100
      COMMON /OUTPUT/ FVT,NODES                                         PRE04110
      DATA BLANK/'          '/                                          PRE04120
      DO 20 I=1,10                                                      PRE04130
20    TIT(I)=BLANK                                                      PRE04140
      DO 21 I=1,3                                                       PRE04150
      DO 21 J=1,NUMNP                                                   PRE04160
21    TFVT(I,J)=0.0D0                                                   PRE04170
      DO 206 KOUNT=1,NPLOAD                                             PRE04180
C                                                                       PRE04190
      NFACE=NNFACE(KOUNT)                                               PRE04200
C     READ ELEMENT NUMBER AND CONNECTIVITY                              PRE04210
C                                                                       PRE04220
      IEL=NPREL(KOUNT)                                                  PRE04230
      DO 19 I=1,20                                                      PRE04240
19    NCON(I)=NCONT(I,IEL)                                              PRE04250
C                                                                       PRE04260
C    READ X,Y,Z COORDINATES OF EACH NODE                                PRE04270
C                                                                       PRE04280
      DO 30 I=1,20                                                      PRE04290
      K=NCON(I)                                                         PRE04300
      COD(I,1)=XPT(K)                                                   PRE04310
      COD(I,2)=YPT(K)                                                   PRE04320
```

```
      COD(I,3)=ZPT(K)                                                   PRE04330
30    CONTINUE                                                          PRE04340
C                                                                       PRE04350
C     READ PRESSURES INPUT AT UP TO 8 NODES,CORNER NODES FIRST          PRE04360
C     THEN MIDSIDE NODES                                                PRE04370
C                                                                       PRE04380
      DO 35 I=1,8                                                       PRE04390
35    PWA(I)=PRESS(I,KOUNT)                                             PRE04400
C                                                                       PRE04410
C                                                                       PRE04420
C     ECHO PRINT ALL INPUT DATA                                         PRE04430
C                                                                       PRE04440
C                                                                       PRE04450
      IF(ICHK2.EQ.1)GO TO 105                                           PRE04460
      WRITE(6,190)                                                      PRE04470
190   FORMAT(///15X,'MATRIX OF NATURAL COORDINATES'///)                 PRE04480
      CALL PRINTD(COD,20,3,2,TIT,20)                                    PRE04490
      WRITE(6,191)                                                      PRE04500
191   FORMAT(///10X,'MATRIX OF CONNECTIVITIES'//)                       PRE04510
      WRITE(6,195)NCON                                                  PRE04520
195   FORMAT(///5X,20I5///)                                             PRE04530
      WRITE(6,192)                                                      PRE04540
192   FORMAT(///10X,'MATRIX OF NODAL PRESSURES'//)                      PRE04550
      CALL PRINTD(PWA,1,8,2,TIT,1)                                      PRE04560
      WRITE(6,100)IEL,NGP,NFACE                                         PRE04570
100   FORMAT(///' THE ELEMENT NUMBER IS ',I4//' THE NUMBER OF GAUSS POINPRE04580
     1TS IS ',I3//' THE FACE TO WHICH PRESSURE IS APPLIED IS  ',I3///)  PRE04590
105   CONTINUE                                                          PRE04600
      CALL CUBAT(NGP)                                                   PRE04610
      IF(ICHK2.EQ.1)GO TO 207                                           PRE04620
      CALL PRINTD(FVT,1,60,1,TIT,1)                                     PRE04630
      SUMX=0.0D0                                                        PRE04640
      SUMY=0.0D0                                                        PRE04650
      SUMZ=0.0D0                                                        PRE04660
      DO 201 I=1,20                                                     PRE04670
201     SUMX=SUMX+FVT(I)                                                PRE04680
      DO 202 I=21,40                                                    PRE04690
202     SUMY=SUMY+FVT(I)                                                PRE04700
      DO 203 I=41,60                                                    PRE04710
203     SUMZ=SUMZ+FVT(I)                                                PRE04720
      WRITE(6,204)SUMX,SUMY,SUMZ                                        PRE04730
204   FORMAT(///' SUMX =',G25.16//' SUMY =',G25.16//' SUMZ =',G25.16//) PRE04740
207   CONTINUE                                                          PRE04750
      DO 205 I=1,8                                                      PRE04760
      K=NODES(I)                                                        PRE04770
      KK=NCON(K)                                                        PRE04780
      TFVT(1,KK)=TFVT(1,KK)+FVT(K)                                      PRE04790
      TFVT(2,KK)=TFVT(2,KK)+FVT(K+20)                                   PRE04800
```

```
      TFVT(3,KK)=TFVT(3,KK)+FVT(K+40)                                   PRE04810
205   CONTINUE                                                          PRE04820
206   CONTINUE                                                          PRE04830
      WRITE(6,209)                                                      PRE04840
209   FORMAT(///5X,'NODE',10X,'X COMPONENT',12X,'Y COMPONENT',15X,       PRE04850
     1'Z COMPONENT'///)                                                 PRE04860
      EPS=1.D-8                                                         PRE04870
      DO 210 NODE=1,NUMNP                                               PRE04880
      A=DABS(TFVT(1,NODE))                                              PRE04890
      B=DABS(TFVT(2,NODE))                                              PRE04900
      C=DABS(TFVT(3,NODE))                                              PRE04910
      IF(A.GT.EPS.OR.B.GT.EPS.OR.C.GT.EPS)WRITE(6,211)NODE,             PRE04920
     1(TFVT(I,NODE),I=1,3)                                              PRE04930
211   FORMAT(1X,I5,4X,3G25.16)                                          PRE04940
210   CONTINUE                                                          PRE04950
      WRITE(6,213)                                                      PRE04960
213   FORMAT(///1X,'NODE',2X,'DOF',1X,'NCUR',12X,'FACTOR'//)            PRE04970
      IOUT=6                                                            PRE04980
      KLOAD=0                                                           PRE04990
214   DO 215 NODE=1,NUMNP                                               PRE05000
      A=DABS(TFVT(1,NODE))                                              PRE05010
      IF(A.GT.EPS)KLOAD=KLOAD+1                                         PRE05020
      AA=TFVT(1,NODE)                                                   PRE05030
      IF(A.GT.EPS)WRITE(IOUT,216)NODE,NCUR,AA                           PRE05040
      B=DABS(TFVT(2,NODE))                                              PRE05050
      IF(B.GT.EPS)KLOAD=KLOAD+1                                         PRE05060
      BB=TFVT(2,NODE)                                                   PRE05070
      IF(B.GT.EPS)WRITE(IOUT,217)NODE,NCUR,BB                           PRE05080
      C=DABS(TFVT(3,NODE))                                              PRE05090
      IF(C.GT.EPS)KLOAD=KLOAD+1                                         PRE05100
      CC=TFVT(3,NODE)                                                   PRE05110
      IF(C.GT.EPS)WRITE(IOUT,218)NODE,NCUR,CC                           PRE05120
215   CONTINUE                                                          PRE05130
      IF(IOUT.EQ.6)WRITE(6,250)KLOAD                                    PRE05140
250   FORMAT(///5X,'THE TOTAL NUMBER OF LOADS IS ',I5///)               PRE05150
216   FORMAT(I5,4X,'1',I5,G25.16)                                       PRE05160
217   FORMAT(I5,4X,'2',I5,G25.16)                                       PRE05170
218   FORMAT(I5,4X,'3',I5,G25.16)                                       PRE05180
      IF(IOUT.EQ.7)STOP                                                 PRE05190
      IF(ICHK3.NE.1)IOUT=7                                              PRE05200
      IF(ICHK3.NE.1)GO TO 214                                           PRE05210
      STOP                                                              PRE05220
      END                                                               PRE05230
      SUBROUTINE CUBAT(NGP)                                             PRE05240
C                                                                       PRE05250
C        PROF GILLES CANTIN AT THE NAVAL POSTGRADUATE SCHOOL SEPT. 1977 PRE05260
C                                                                       PRE05270
      IMPLICIT REAL*8(A-H,O-Z)                                          PRE05280
```

```
      DIMENSION COD(20,3), PWA(8),GP(6,5),FVT(60)    ,WF(6,5),NCON(20)  PRE05290
      COMMON/PRESR/PWA,WFAC,NFACE                                       PRE05300
      COMMON/CHECK/ICHK1,ICHK2,ICHK3,ICHK4                              PRE05310
      COMMON /OUTPUT/ FVT,NODES                                         PRE05320
      DATA GP/.57735026918963D0,-.57735026918963DC,0.0D0,0.0D0,0.0D0,0.DPRE05330
     10,.77459666924148D0,0.0D0,-.77459666924148D0,0.0D0,0.0D0,0.0D0,.86PRE05340
     2113631159405D0,0.33998104358486D0,-.33998104358486D0,-.86113631159PRE05350
     3405D0,0.0D0,0.0D0,.90617984593866D0,-.53846931010568DC,0.0D0,-.5384PRE05360
     46931010568D0,-.90617984593866D0, 0.0D0,.93246951420315D0,.6612093PRE05370
     5646626D0,.23861918608317D0,-.23861918608317D0,-.66120938646626D0,-PRE05380
     6.93246951420315D0/                                                PRE05390
      DATA WF/1.0D0,1.0D0,0.0D0,0.0D0,0.0D0,0.0D0,.555555555555556DC,.888PRE05400
     1888888888889D0,.555555555555556D0,0.0D0,0.0D0,0.0D0,.34785484513745DPRE05410
     20,.65214515486255D0,.65214515486255D0,.34785484513745D0,0.0D0,0.0DPRE05420
     30,.23692688505619D0,.47862867049937D0,.56888888888889D0,.47862867PRE05430
     449937D0,.23692688505619D0,0.0D0,.17132449237917D0,.36076157304814DPRE05440
     50,.46791393457269D0,.46791393457269D0,.36076157304814D0,.17132449PRE05450
     637917D0/                                                          PRE05460
      DO 100 I=1,60                                                     PRE05470
100   FVT(I)=0.0D0                                                      PRE05480
      IF(NGP.LE.0.OR.NGP.GT.6) NGP=3                                    PRE05490
      ZRO =0.0D0                                                        PRE05500
      IDGP = NGP-1                                                      PRE05510
      WR=1.0D0                                                          PRE05520
      WS=1.0D0                                                          PRE05530
      WT=1.0D0                                                          PRE05540
      NGP1=NGP                                                          PRE05550
      NGP2=NGP                                                          PRE05560
      NGP3=NGP                                                          PRE05570
      IF(NFACE.EQ.1.OR.NFACE.EQ.2)NGP1=1                                PRE05580
      IF(NFACE.EQ.3.OR.NFACE.EQ.4)NGP2=1                                PRE05590
      IF(NFACE.EQ.5.OR.NFACE.EQ.6)NGP3=1                                PRE05600
      DO 200 I=1,NGP1                                                   PRE05610
      IF(NFACE.EQ.1)R=1.0D0                                             PRE05620
      IF(NFACE.EQ.2)R=-1.0D0                                            PRE05630
      IF(NFACE.EQ.1.OR.NFACE.EQ.2)GO TO 120                             PRE05640
      R = GP(I,IDGP)                                                    PRE05650
      WR= WF(I,IDGP)                                                    PRE05660
120   CONTINUE                                                          PRE05670
      DO 200 J=1,NGP2                                                   PRE05680
      IF(NFACE.EQ.3)S=1.0D0                                             PRE05690
      IF(NFACE.EQ.4)S=-1.0D0                                            PRE05700
      IF(NFACE.EQ.3.OR.NFACE.EQ.4)GO TO 130                             PRE05710
      S = GP(J,IDGP)                                                    PRE05720
      WS= WF(J,IDGP)                                                    PRE05730
130   CONTINUE                                                          PRE05740
      DO 200 K=1,NGP3                                                   PRE05750
      IF(NFACE.EQ.5)T=1.0D0                                             PRE05760
```

```
      IF(NFACE.EQ.6)T=-1.0D0                                                PRE05770
      IF(NFACE.EQ.5.OR.NFACE.EQ.6)GO TO 140                                 PRE05780
      T = GP(K,IDGP)                                                        PRE05790
      WT= WF(K,IDGP)                                                        PRE05800
140   CONTINUE                                                              PRE05810
      WFAC = WR*WS*WT                                                       PRE05820
      IF(ICHK2.EQ.1)GO TO 145                                               PRE05830
      WRITE(6,160)R,S,T,WFAC                                                PRE05840
145   CONTINUE                                                              PRE05850
      CALL SHP20(R,S,T)                                                     PRE05860
  200 CONTINUE                                                              PRE05870
160   FORMAT(1X,4G25.16)                                                    PRE05880
      RETURN                                                                PRE05890
      END                                                                   PRE05900
      SUBROUTINE SHP20(R,S,T)                                               PRE05910
C                                                                           PRE05920
C        UPON EXIT FROM THE PRESENT SUBROUTINE, FVT WILL CONTAIN            PRE05930
C        THE VALUES OF LOAD VECTOR ARGUMENT FVT(20,3) THAT CAN BE           PRE05940
C        USED FOR GAUSSIAN QUADRATURE                                       PRE05950
C                                                                           PRE05960
C        THE MATRIX ARGUMENT IS THE RESULT OF THE PRODUCT OF THE            PRE05970
C        FOLLOWING MATRICES:  NT(20,3)*A(3,3)*N(3,20)*XYZ(20,3)*DETJ        PRE05980
C        WHERE THE MATRIX N(3,20) IS THE MATRIX OF SHAPE FUNCTIONS          PRE05990
C        NT(20,3) IS THE TRANSPOSE OF N(3,20), XYZ(20,3) IS THE             PRE06000
C        MATRIX OF COORDINATES OF THE NODES AND DETJ IS THE DETERMINANT     PRE06010
C        OF THE JACOBIAN MATRIX                                             PRE06020
C                                                                           PRE06030
C        R,S,T ARE THE NON DIMENSIONAL COORDINATES OF THE ELEMENT           PRE06040
C        DTJ IS THE DETERMINANT OF THE JACOBIAN MATRIX                      PRE06050
C        FVT  IS THE OUTPUT VECTOR                                          PRE06060
C        COD  IS AN INPUT ARRAY OF THE ELEMENT COORDINATES IN THE           PRE06070
C        GLOBAL COORDINAT SYSTEM                                            PRE06080
C        NCON IS THE CONNECTIVITY MATRIX OF THE ELEMENT                     PRE06090
C        IEL  IS THE ELEMENT IDENTIFICATION NUMBER                          PRE06100
C                                                                           PRE06110
C        PROF. GILLES CANTIN  SEPT. 1977  NAVAL POSTGRADUATE SCHOOL         PRE06120
C        MONTEREY CALIFORNIA  93940                                         PRE06130
C                                                                           PRE06140
      IMPLICIT REAL*8(A-H,O-Z)                                              PRE06150
      DIMENSION COD(20,3),CNI(3,20),EMI(20)                                 PRE06160
      DIMENSION NCON(20)                                                    PRE06170
      REAL*8 JAC(3,3)                                                       PRE06180
      COMMON/GEOM/COD,NCON,IEL                                              PRE06190
      GP(BT) = 0.5D0*(1.0D0 + BT)                                           PRE06200
      GM(BT) = 0.5D0*(1.0D0 - BT)                                           PRE06210
      GZ(BT) = 1.0D0 - BT*BT                                                PRE06220
      DP(BT) =  0.5D0+ZRO*BT                                                PRE06230
      DM(BT) = -0.5D0+ZRO*BT                                                PRE06240
```

```
      DZ(BT) = -2.0D0*BT                                                                                                PRE06250
      ZRO = 0.0D0                                                                                                       PRE06260
      ONE = 1.0D0                                                                                                       PRE06270
      TWO = 2.0D0                                                                                                       PRE06280
      DO 100 I=1,3                                                                                                      PRE06290
      DO 100 J=1,20                                                                                                     PRE06300
  100 DNI(I,J) = ZRO                                                                                                    PRE06310
      EMI( 9) = GZ(R)*GP(S)*GP(T)                                                                                       PRE06320
      IF(NCON( 9).EQ.0) EMI( 9) = ZRO                                                                                   PRE06330
      EMI(10) = GM(R)*GZ(S)*GP(T)                                                                                       PRE06340
      IF(NCON(10).EQ.0) EMI(10) = ZRO                                                                                   PRE06350
      EMI(11) = GZ(R)*GM(S)*GP(T)                                                                                       PRE06360
      IF(NCON(11).EQ.0) EMI(11) = ZRO                                                                                   PRE06370
      EMI(12) = GP(R)*GZ(S)*GP(T)                                                                                       PRE06380
      IF(NCON(12).EQ.0) EMI(12) = ZRO                                                                                   PRE06390
      EMI(13) = GZ(R)*GP(S)*GM(T)                                                                                       PRE06400
      IF(NCON(13).EQ.0) EMI(13) = ZRO                                                                                   PRE06410
      EMI(14) = GM(R)*GZ(S)*GM(T)                                                                                       PRE06420
      IF(NCON(14).EQ.0) EMI(14) = ZRO                                                                                   PRE06430
      EMI(15) = GZ(R)*GM(S)*GM(T)                                                                                       PRE06440
      IF(NCON(15).EQ.0) EMI(15) = ZRO                                                                                   PRE06450
      EMI(16) = GP(R)*GZ(S)*GM(T)                                                                                       PRE06460
      IF(NCON(16).EQ.0) EMI(16) = ZRO                                                                                   PRE06470
      EMI(17) = GP(R)*GP(S)*GZ(T)                                                                                       PRE06480
      IF(NCON(17).EQ.0) EMI(17) = ZRO                                                                                   PRE06490
      EMI(18) = GM(R)*GP(S)*GZ(T)                                                                                       PRE06500
      IF(NCON(18).EQ.0) EMI(18) = ZRO                                                                                   PRE06510
      EMI(19) = GM(R)*GM(S)*GZ(T)                                                                                       PRE06520
      IF(NCON(19).EQ.0) EMI(19) = ZRO                                                                                   PRE06530
      EMI(20) = GP(R)*GM(S)*GZ(T)                                                                                       PRE06540
      IF(NCON(20).EQ.0) EMI(20) = ZRO                                                                                   PRE06550
      EMI( 1) = GP(R)*GP(S)*GP(T) - (EMI( 9) + EMI(12) + EMI(17))/TWO                                                   PRE06560
      EMI( 2) = GM(R)*GP(S)*GP(T) - (EMI( 9) + EMI(10) + EMI(18))/TWO                                                   PRE06570
      EMI( 3) = GM(R)*GM(S)*GP(T) - (EMI(10) + EMI(11) + EMI(19))/TWO                                                   PRE06580
      EMI( 4) = GP(R)*GM(S)*GP(T) - (EMI(11) + EMI(12) + EMI(20))/TWO                                                   PRE06590
      EMI( 5) = GP(R)*GP(S)*GM(T) - (EMI(13) + EMI(16) + EMI(17))/TWO                                                   PRE06600
      EMI( 6) = GM(R)*GP(S)*GM(T) - (EMI(13) + EMI(14) + EMI(18))/TWO                                                   PRE06610
      EMI( 7) = GM(R)*GM(S)*GM(T) - (EMI(14) + EMI(15) + EMI(19))/TWO                                                   PRE06620
      EMI( 8) = GP(R)*GM(S)*GM(T) - (EMI(15) + EMI(16) + EMI(20))/TWO                                                   PRE06630
      IF(NCON( 9).EQ.0) GO TO 500                                                                                       PRE06640
      DNI(1, 9) = DZ(R)*GP(S)*GP(T)                                                                                     PRE06650
      DNI(2, 9) = GZ(R)*DP(S)*GP(T)                                                                                     PRE06660
      DNI(3, 9) = GZ(R)*GP(S)*DP(T)                                                                                     PRE06670
  500 IF(NCON(10).EQ.0) GO TO 510                                                                                       PRE06680
      DNI(1,10) = DM(R)*GZ(S)*GP(T)                                                                                     PRE06690
      DNI(2,10) = GM(R)*DZ(S)*GP(T)                                                                                     PRE06700
      DNI(3,10) = GM(R)*GZ(S)*DP(T)                                                                                     PRE06710
  510 IF(NCON(11).EQ.0) GO TO 520                                                                                       PRE06720
```

```
      DNI(1,11) = DZ(R)*GM(S)*GP(T)                                                  PRE06730
      DNI(2,11) = GZ(R)*DM(S)*GP(T)                                                  PRE06740
      DNI(3,11) = GZ(R)*GM(S)*DP(T)                                                  PRE06750
  520 IF(NCON(12).EQ.0) GO TO 530                                                    PRE06760
      DNI(1,12) = DP(R)*GZ(S)*GP(T)                                                  PRE06770
      DNI(2,12) = GP(R)*DZ(S)*GP(T)                                                  PRE06780
      DNI(3,12) = GP(R)*GZ(S)*DP(T)                                                  PRE06790
  530 IF(NCON(13).EQ.0) GO TO 540                                                    PRE06800
      DNI(1,13) = DZ(R)*GP(S)*GM(T)                                                  PRE06810
      DNI(2,13) = GZ(R)*DP(S)*GM(T)                                                  PRE06820
      DNI(3,13) = GZ(R)*GP(S)*DM(T)                                                  PRE06830
  540 IF(NCON(14).EQ.0) GO TO 550                                                    PRE06840
      DNI(1,14) = DM(R)*GZ(S)*GM(T)                                                  PRE06850
      DNI(2,14) = GM(R)*DZ(S)*GM(T)                                                  PRE06860
      DNI(3,14) = GM(R)*GZ(S)*DM(T)                                                  PRE06870
  550 IF(NCON(15).EQ.0) GO TO 560                                                    PRE06880
      DNI(1,15) = DZ(R)*GM(S)*GM(T)                                                  PRE06890
      DNI(2,15) = GZ(R)*DM(S)*GM(T)                                                  PRE06900
      DNI(3,15) = GZ(R)*GM(S)*DM(T)                                                  PRE06910
  560 IF(NCON(16).EQ.0) GO TO 570                                                    PRE06920
      DNI(1,16) = DP(R)*GZ(S)*GM(T)                                                  PRE06930
      DNI(2,16) = GP(R)*DZ(S)*GM(T)                                                  PRE06940
      DNI(3,16) = GP(R)*GZ(S)*DM(T)                                                  PRE06950
  570 IF(NCON(17).EQ.0) GO TO 580                                                    PRE06960
      DNI(1,17) = DP(R)*GP(S)*GZ(T)                                                  PRE06970
      DNI(2,17) = GP(R)*DP(S)*GZ(T)                                                  PRE06980
      DNI(3,17) = GP(R)*GP(S)*DZ(T)                                                  PRE06990
  580 IF(NCON(18).EQ.0) GO TO 590                                                    PRE07000
      DNI(1,18) = DM(R)*GP(S)*GZ(T)                                                  PRE07010
      DNI(2,18) = GM(R)*DP(S)*GZ(T)                                                  PRE07020
      DNI(3,18) = GM(R)*GP(S)*DZ(T)                                                  PRE07030
  590 IF(NCON(19).EQ.0) GO TO 600                                                    PRE07040
      DNI(1,19) = DM(R)*GM(S)*GZ(T)                                                  PRE07050
      DNI(2,19) = GM(R)*DM(S)*GZ(T)                                                  PRE07060
      DNI(3,19) = GM(R)*GM(S)*DZ(T)                                                  PRE07070
  600 IF(NCON(20).EQ.0) GO TO 620                                                    PRE07080
      DNI(1,20) = DP(R)*GM(S)*GZ(T)                                                  PRE07090
      DNI(2,20) = GP(R)*DM(S)*GZ(T)                                                  PRE07100
      DNI(3,20) = GP(R)*GM(S)*DZ(T)                                                  PRE07110
  620 CONTINUE                                                                       PRE07120
      DNI(1,1) = DP(R)*GP(S)*GP(T) - (DNI(1, 9)+DNI(1,12)+DNI(1,17))/TWOPRE07130
      DNI(2,1) = GP(R)*DP(S)*GP(T) - (DNI(2, 9)+DNI(2,12)+DNI(2,17))/TWOPRE07140
      DNI(3,1) = GP(R)*GP(S)*DP(T) - (DNI(3, 9)+DNI(3,12)+DNI(3,17))/TWOPRE07150
      DNI(1,2) = DM(R)*GP(S)*GP(T) - (DNI(1, 9)+DNI(1,10)+DNI(1,18))/TWOPRE07160
      DNI(2,2) = GM(R)*DP(S)*GP(T) - (DNI(2, 9)+DNI(2,10)+DNI(2,18))/TWOPRE07170
      DNI(3,2) = GM(R)*GP(S)*DP(T) - (DNI(3, 9)+DNI(3,10)+DNI(3,18))/TWOPRE07180
      DNI(1,3) = DM(R)*GM(S)*GP(T) - (DNI(1,10)+DNI(1,11)+DNI(1,19))/TWOPRE07190
      DNI(2,3) = GM(R)*DM(S)*GP(T) - (DNI(2,10)+DNI(2,11)+DNI(2,19))/TWOPRE07200
```

```
      DNI(3,3) = GM(R)*GM(S)*DP(T) - (DNI(3,10)+DNI(3,11)+DNI(3,19))/TWOPRE07210
      DNI(1,4) = DP(R)*GM(S)*GP(T) - (DNI(1,11)+DNI(1,12)+DNI(1,20))/TWOPRE07220
      DNI(2,4) = GP(R)*DM(S)*GP(T) - (DNI(2,11)+DNI(2,12)+DNI(2,20))/TWOPRE07230
      DNI(3,4) = GP(R)*GM(S)*DP(T) - (DNI(3,11)+DNI(3,12)+DNI(3,20))/TWOPRE07240
      DNI(1,5) = DP(R)*GP(S)*GM(T) - (DNI(1,13)+DNI(1,16)+DNI(1,17))/TWOPRE07250
      DNI(2,5) = GP(R)*DP(S)*GM(T) - (DNI(2,13)+DNI(2,16)+DNI(2,17))/TWOPRE07260
      DNI(3,5) = GP(R)*GP(S)*DM(T) - (DNI(3,13)+DNI(3,16)+DNI(3,17))/TWOPRE07270
      DNI(1,6) = DM(R)*GP(S)*GM(T) - (DNI(1,13)+DNI(1,14)+DNI(1,18))/TWOPRE07280
      DNI(2,6) = GM(R)*DP(S)*GM(T) - (DNI(2,13)+DNI(2,14)+DNI(2,18))/TWOPRE07290
      DNI(3,6) = GM(R)*GP(S)*DM(T) - (DNI(3,13)+DNI(3,14)+DNI(3,18))/TWOPRE07300
      DNI(1,7) = DM(R)*GM(S)*GM(T) - (DNI(1,14)+DNI(1,15)+DNI(1,19))/TWOPRE07310
      DNI(2,7) = GM(R)*DM(S)*GM(T) - (DNI(2,14)+DNI(2,15)+DNI(2,19))/TWOPRE07320
      DNI(3,7) = GM(R)*GM(S)*DM(T) - (DNI(3,14)+DNI(3,15)+DNI(3,19))/TWOPRE07330
      DNI(1,8) = DP(R)*GM(S)*GM(T) - (DNI(1,15)+DNI(1,16)+DNI(1,20))/TWOPRE07340
      DNI(2,8) = GP(R)*DM(S)*GM(T) - (DNI(2,15)+DNI(2,16)+DNI(2,20))/TWOPRE07350
      DNI(3,8) = GP(R)*GM(S)*DM(T) - (DNI(3,15)+DNI(3,16)+DNI(3,20))/TWOPRE07360
      DO 630 I=1,3                                                    PRE07370
      DO 630 J=1,3                                                    PRE07380
      JAC(I,J)=ZRO                                                    PRE07390
      DO 630 K=1,20                                                   PRE07400
  630 JAC(I,J) = JAC(I,J) + DNI(I,K)*COO(K,J)                         PRE07410
      A1 = JAC(1,1)*JAC(2,2)*JAC(3,3)                                 PRE07420
      A2 = JAC(1,2)*JAC(2,3)*JAC(3,1)                                 PRE07430
      A3 = JAC(1,3)*JAC(2,1)*JAC(3,2)                                 PRE07440
      B1 = JAC(3,1)*JAC(2,2)*JAC(1,3)                                 PRE07450
      B2 = JAC(3,2)*JAC(2,3)*JAC(1,1)                                 PRE07460
      B3 = JAC(3,3)*JAC(2,1)*JAC(1,2)                                 PRE07470
      DTJ = A1+A2+A3 - (B1+B2+B3)                                     PRE07480
      CALL FACEP(EMI,DNI,JAC)                                         PRE07490
      IF(DTJ.LE.ZRO) GO TO 960                                        PRE07500
      RETURN                                                          PRE07510
  960 WRITE(6,1000)IEL,DTJ,R,S,T                                      PRE07520
 1000 FORMAT(//,5X,'ELEMENT NUMBER ',I5,' HAS A JACOBIAN DETERMINANT',/ PRE07530
     1,5X,'OF VALUE ',1G25.16,'*** EXECUTION STOPS ***'/,             PRE07540
     2 5X,' R = ',G25.16/5X,' S = ',G25.16/5X,' T = ',G25.16//)       PRE07550
      STOP                                                            PRE07560
      END                                                             PRE07570
      SUBROUTINE FACEP(EMI,DNI,JAC)                                   PRE07580
      IMPLICIT REAL*8(A-H,O-Z)                                        PRE07590
      DIMENSION NODES(8),KFACE(6,8),FVAL(6),PR(8),P(20)               PRE07600
      DIMENSION PWA(8),EMI(20),DNI(3,20),DUM(60),FVT(60)              PRE07610
      DIMENSION BIGN(3,60),PROD1(60)                                  PRE07620
      REAL*8 JAC(3,3),NORML(3)                                        PRE07630
      COMMON/PRESR/PWA,WFAC,NFACE                                     PRE07640
      COMMON /OUTPUT/ FVT,NODES                                       PRE07650
      DATA KFACE / 1,2,1,4,1,5,                                       PRE07660
     1             4,3,5,8,2,6,                                       PRE07670
     2             8,7,6,7,3,7,                                       PRE07680
```

```
     3              5,6,2,3,4,8,                                          PRE07690
     4              12,10,17,20,9,13,                                     PRE07700
     5              20,19,13,15,10,14,                                    PRE07710
     6              16,14,18,19,11,15,                                    PRE07720
     7              17,18,9,11,12,16/                                     PRE07730
      DATA FVAL /1.0D0,-1.0D0,1.0D0,-1.0D0,1.0D0,-1.0D0/                  PRE07740
C                                                                         PRE07750
CCC   DETERMINE ELEMENT NODES CONTRIBUTING TO FORCE CALCULATIONS          PRE07760
C     ON THIS FACE                                                        PRE07770
C                                                                         PRE07780
      DO 2 I=1,8                                                          PRE07790
      NODES(I)=KFACE(NFACE,I)                                             PRE07800
2     CONTINUE                                                            PRE07810
C                                                                         PRE07820
CCC   SET UP PRESSURE VECTOR FOR UP TO EIGHT FACE NODES                   PRE07830
CCC                                                                       PRE07840
CCC         1. ADJUST THE SIGN OF THE PRESSURES SO THE POSITIVE           PRE07850
CCC          PRESSURES ALWAYS TENSION THE ELEMENT                         PRE07860
CCC                                                                       PRE07870
C                                                                         PRE07880
      FACT=-FVAL(NFACE)                                                   PRE07890
C                                                                         PRE07900
CCC         2. DISTRIBUTED PRESSURES GIVEN AT UP TO EIGHT NODES           PRE07910
CCC           ON THE FACE                                                 PRE07920
C                                                                         PRE07930
      DO 25 K=1,8                                                         PRE07940
C     PWA(I) = VALUE OF PRESSURE AT THE ITH NODE                          PRE07950
      PR(K)=PWA(K)*FACT                                                   PRE07960
25    CONTINUE                                                            PRE07970
C                                                                         PRE07980
C     COMPUTE THE DIRECTION NUMBERS: NORML(1),NORML(2),NORML(3)           PRE07990
C                                                                         PRE08000
      IF(NFACE.GT.2)GO TO 60                                              PRE08010
      NORML(1)=JAC(2,2)*JAC(3,3)-JAC(2,3)*JAC(3,2)                        PRE08020
      NORML(2)=JAC(2,3)*JAC(3,1)-JAC(2,1)*JAC(3,3)                        PRE08030
50    NORML(3)=JAC(2,1)*JAC(3,2)-JAC(2,2)*JAC(3,1)                        PRE08040
      GO TO 94                                                            PRE08050
60    IF(NFACE.GT.4)GO TO 80                                              PRE08060
      NORML(1)=JAC(3,2)*JAC(1,3)-JAC(3,3)*JAC(1,2)                        PRE08070
      NORML(2)=JAC(3,3)*JAC(1,1)-JAC(3,1)*JAC(1,3)                        PRE08080
70    NORML(3)=JAC(3,1)*JAC(1,2)-JAC(3,2)*JAC(1,1)                        PRE08090
      GO TO 94                                                            PRE08100
80    CONTINUE                                                            PRE08110
      NORML(1)=JAC(1,2)*JAC(2,3)-JAC(1,3)*JAC(2,2)                        PRE08120
      NORML(2)=JAC(1,3)*JAC(2,1)-JAC(1,1)*JAC(2,3)                        PRE08130
90    NORML(3)=JAC(1,1)*JAC(2,2)-JAC(1,2)*JAC(2,1)                        PRE08140
94    CONTINUE                                                            PRE08150
      IF(NFACE.EQ.1.OR.NFACE.EQ.3.OR.NFACE.EQ.5)GO TO 100                 PRE08160
```

```
      DO 95 I=1,3                                                       PRE08170
95    NORML(I)=-1.0D0*NORML(I)                                          PRE08180
100   CONTINUE                                                          PRE08190
C                                                                       PRE08200
C     ASSEMBLE LARGE SHAPE FACTOR MATRIX: BIGN(3,60)                    PRE08210
C                                                                       PRE08220
C     ZERO ALL ELEMENTS OF BIGN                                         PRE08230
      DO 105 I=1,3                                                      PRE08240
      DO 110 J=1,60                                                     PRE08250
      BIGN(I,J)=0.0D0                                                   PRE08260
110   CONTINUE                                                          PRE08270
105   CONTINUE                                                          PRE08280
      DO 120 J=1,20                                                     PRE08290
120   BIGN(1,J)=EMI(J)                                                  PRE08300
      DO 130 J=21,40                                                    PRE08310
      K=J-20                                                            PRE08320
130   BIGN(2,J)=EMI(K)                                                  PRE08330
      DO 140 J=41,60                                                    PRE08340
      K=J-40                                                            PRE08350
140   BIGN(3,J)=EMI(K)                                                  PRE08360
C                                                                       PRE08370
C     ASSEMBLE PRESSURE MATRIX(3,20)                                    PRE08380
C                                                                       PRE08390
C     ZERO ALL ELEMENTS OF P MATRIX                                     PRE08400
C                                                                       PRE08410
      DO 150 I=1,20                                                     PRE08420
150   P(I)=0.0D0                                                        PRE08430
      DO 160 I=1,8                                                      PRE08440
      J=NODES(I)                                                        PRE08450
160   P(J)=PWA(I)                                                       PRE08460
C                                                                       PRE08470
C     PERFORM MATRIX MULTIPLICATION: NORML(1,3)*BIGN(3,60)*NT(1,20)     PRE08480
C     *P(20,1)                                                          PRE08490
C  PROD1(1,60)=NORML*BIGN; PROD2(1)=NT*P                                PRE08500
C                                                                       PRE08510
      DO 170 J=1,60                                                     PRE08520
      PROD1(J)=0.0D0                                                    PRE08530
      DO 170 K=1,3                                                      PRE08540
170   PROD1(J)=PROD1(J)+NORML(K)*BIGN(K,J)                              PRE08550
      PROD2=0.0D0                                                       PRE08560
      DO 180 I=1,20                                                     PRE08570
180   PROD2=PROD2+EMI(I)*P(I)                                           PRE08580
C                                                                       PRE08590
C     PERFORM MULTIPLICATION OF PROD1*PROD2                             PRE08600
C                                                                       PRE08610
      DO 190 J=1,60                                                     PRE08620
190   DUM(J)=PROD1(J)*PROD2                                             PRE08630
C                                                                       PRE08640
```

```
C      PERFORM INTEGRATION                                                      PRE08650
C                                                                               PRE08660
       DO 195 I=1,60                                                            PRE08670
195    FVT(I)=FVT(I)+DUM(I)*WFAC                                                PRE08680
       RETURN                                                                   PRE08690
       END                                                                      PRE08700
       SUBROUTINE PRINTS(A,N,M,ICODE,TITLE,ND)                                  PRE08710
C*****************************************************************************PRE08720
C                                                                               PRE08730
CC     THIS SUBROUTINE WILL PRINT ANY MATRIX IN A CONVENIENT AND                PRE08740
CC     EASILY READABLE FORMAT.                                                  PRE08750
CC                                                                              PRE08760
CC     FOR SINGLE PRECISION ARRAYS USE THE FOLLOWING CALL:                      PRE08770
CC         CALL PRINTS(A,N,M,ICODE,TITLE,ND)                                    PRE08780
CC     FOR DOUBLE PRECISION ARRAYS USE:                                         PRE08790
CC         CALL PRINTD(A,N,M,ICODE,TITLE,ND)                                    PRE08800
CC                                                                              PRE08810
CC           A IS THE NAME OF THE ARRAY TO BE PRINTED                           PRE08820
CC           N IS THE NUMBER OF ROWS TO BE PRINTED                              PRE08830
CC           M IS THE NUMBER OF COLUMNS TO BE PRINTED                           PRE08840
CC       ICODE   1  FOR  4  COLUMNS PER PAGE (THE MATRIX                        PRE08850
CC               2   -   6     -        -          IS PARTITIONED              PRE08860
CC               3   -   8     -        -               IN VERTICAL            PRE08870
CC               4   -  10     -        -                  BLOCKS)             PRE08880
CC               5   -  12     -        -                                      PRE08890
CC               (FOR SINGLE PRECISION,  ICODE  VALUES OF 1 AND 2               PRE08900
CC               SHOULD BE AVOIDED)                                             PRE08910
CC       TITLE IS A DOUBLE PRECISION ARRAY DIMENSIONED (10), CONTAINING         PRE08920
CC             ANY TITLE                                                        PRE08930
CC          ND IS THE ACTUAL ROW DIMENSION OF THE ARRAY A IN THE CALLING        PRE08940
CC             PROGRAM                                                          PRE08950
CC                                                                              PRE08960
CC     PROF. GILLES CANTIN  NAVAL POSTGRADUATE SCHOOL 1970                      PRE08970
C                                                                               PRE08980
C*****************************************************************************PRE08990
       DIMENSION A(ND,1),NCOL(5)                                                PRE09000
       REAL*8 TITLE(10),F(4,5),FM1(2),FM2(2)                                    PRE09010
       DATA NCOL/4,6,8,10,12/                                                   PRE09020
       DATA  F/'(/X4I25)',' ',' ','(I5,2X1P',',4G25.16)',                       PRE09030
      2        '(/X6I18)',' ',' ','(I5,2X1P',',6G18.10)',                      PRE09040
      3        '(/2X8I15',',')  ',','(I5,2X1P',',8G15.7)  ',                  PRE09050
      4        '(/3X10I11',',2)  ',','(I5,2X1P',',10G12.4)',                  PRE09060
      5        '(/4X12I11',',0)  ',','(I5,2X1P',',12G10.2)'/                  PRE09070
       IF(ICODE.LT.1.OR.ICODE.GT.5) ICODE=4                                     PRE09080
       NK=NCOL(ICODE)                                                           PRE09090
       FM1(1)=F(1,ICODE)                                                        PRE09100
       FM1(2)=F(2,ICODE)                                                        PRE09110
       FM2(1)=F(3,ICODE)                                                        PRE09120
```

```
      FM2(2)=F(4,ICODE)                                                 PRE09130
      NPR = 1 + (M-1)/NK                                                PRE09140
      N1 = 1                                                            PRE09150
      N2 = MINO(M,NK)                                                   PRE09160
      WRITE(6,1000) TITLE                                               PRE09170
 1000 FORMAT(//10A8/)                                                   PRE09180
      DO 200 K=1,NPR                                                    PRE09190
      WRITE(6,FM1) (J,J=N1,N2)                                          PRE09200
      DO 100 I=1,N                                                      PRE09210
  100 WRITE(6,FM2) I,(A(I,J),J=N1,N2)                                   PRE09220
      N1 = N1 + NK                                                      PRE09230
  200 N2 = MINO(M,N2+NK)                                                PRE09240
      RETURN                                                            PRE09250
      ENTRY       PRINTD(B,N,M,ICODE,TITLE,ND)                          PRE09260
      DIMENSION B(ND,1)                                                 PRE09270
      REAL*8 B                                                          PRE09280
      IF(ICODE.LT.1.OR.ICODE.GT.5) ICODE=4                              PRE09290
      NK=NCOL(ICODE)                                                    PRE09300
      FM1(1)=F(1,ICODE)                                                 PRE09310
      FM1(2)=F(2,ICODE)                                                 PRE09320
      FM2(1)=F(3,ICODE)                                                 PRE09330
      FM2(2)=F(4,ICODE)                                                 PRE09340
      NPR = 1 + (M-1)/NK                                                PRE09350
      N1 = 1                                                            PRE09360
      N2 = MINO(M,NK)                                                   PRE09370
      WRITE(6,1000) TITLE                                               PRE09380
      DO 210 K=1,NPR                                                    PRE09390
      WRITE(6,FM1) (J,J=N1,N2)                                          PRE09400
      DO 110 I=1,N                                                      PRE09410
  110 WRITE(6,FM2) I,(B(I,J),J=N1,N2)                                   PRE09420
      N1 = N1 + NK                                                      PRE09430
  210 N2 = MINO(M,N2+NK)                                                PRE09440
      RETURN                                                            PRE09450
      END                                                               PRE09460
```

LIST OF REFERENCES

1. Admiralty Materials Laboratory Report No. 1/75, The Properties of Silicon Nitride Ceramics and their Use in Service Thermal Environments, by D. J. Godfrey, February 1975.

2. Tree, D. J. et al, "Three Dimensional Optimization of a Gas Turbine Disk and Blade Attachment", Journal of Aircraft, Vol. 13, p. 994-999, December 1976.

3. AiResearch Manufacturing Company of Arizona Report 76-212409, Documentation of Ceramic Component Design Analyses - Ceramic Gas Turbine Engine Demonstration Program, Appendix II, 1 December 1976.

4. Massachusetts Institute of Technology Report 82448-1, A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis, by K. J. Bathe, September 1975.

5. Earthquake Engineering Research Center Report EERC 73-11, SAP IV - A Structural Analysis Program for Static and Dynamic Response of Linear Systems, by K. J. Bathe, E. L. Wilson, F. E. Peterson, June 1973.

6. AiResearch Manufacturing Company of Arizona Report 76-212188(3), Ceramic Gas Turbine Engine Demonstration Interim Report Number 3, p. 24-30, December 1976.

7. Roark, R. J. and Young, W. C., Formulas for Stress and Strain, Fifth ed., Table 3, McGraw-Hill Publishing Company, 1975.

8. Bathe, K. J. and Wilson, E. L., Numerical Methods in Finite Element Analysis, Prentice Hall Inc., 1976.

9. Boley, B. A., and Weiner, J. H., Theory of Thermal Stresses, John Wiley and Sons, Inc., 1960.

10. AiResearch Manufacturing Company of Arizona, TL 310-1936, Sheets 1 through 10, Blade Sections Rotor First Stage Turbine, 20 September 1976.

11. Kibler, A. E., A Finite Element Preprocessor for SAP IV and ADINA, M.S. Thesis, Naval Postgraduate School, Monterey, CA., September 1977.

12. Easterling, L. R., Stress Analysis of Ceramic Gas Turbine Blades by the Finite Element Method - Part I, M.E. Thesis, Naval Postgraduate School, Monterey, CA, March 1978.

13. AiResearch Manufacturing Company of Arizona Report 76-212188(2), Ceramic Gas Turbine Engine Demonstration Interim Report Number 3, p. 2.5 - 7, September 1976.

14. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill Publishing Company, 1971.

15. W. R. Church Computer Center, Naval Postgraduate School, Monterey, CA, User's Manual, March 1970.

16. Losh, D. M., The Implementation of a Finite Element Computer Code and Associated Pre- and Postprocessor into AE 4101 and 4102 (Flight Vehicle Structural Analysis I and II), M.S. Thesis, Naval Postgraduate School, Monterey, CA, December 1976.

17. Naval Postgraduate School Technical Note TN No. 0141-05, User Libraries and Source Code Editing Under OS, by Sharon D. Raney, Revised July 1977.

18. Timoshenko, S. and Goodier, J. N., Theory of Elasticity, p. 254, 2nd ed., McGraw-Hill Book Company, 1951.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor Gilles Cantin, Code 69Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	12
5.	LT Adrian E. Kibler Jr., USN Third Ave. Hastings, Pennsylvania 16646	1
6.	Prof. K. J. Bathe Mechanical Engineering Department M.I.T. 77 Massachusetts Avenue Cambridge, Massachusetts 02139	2
7.	Jack Tree AiResearch Manufacturing Company 402 South 36th Street P.O. Box 5217 Phoenix, Arizona 85010	1
8.	Prof. Edward L. Wilson Structural Engineering Division Civil Engineering Department University of California (Berkeley) Berkeley, California 94720	1
9.	LT. John H. Preisel, Jr., USN 922 Bernard Rd Peekskill, New York 10566	2
10.	LCDR. Lael R. Easterling, USN 4243 N.W. 54th Oklahoma City, Oklahoma 73112	1

11. Dr. Jean Louis Batoz, (I-230) 1
Department of Civil Engineering
M.I.T.
Cambridge, Massachusetts 02139

12. Prof. Guri Dhatt 1
Centre Technique de l'Informatique
Universite Laval
Quebec, Prov. de Quebec
Canada, G1K7P4

13. John Fairbanks 1
Department of Energy
Division of Power Systems
20 Massachusetts Ave. N.W.
Washington, D. C. 20545

14. Dr. Gilbert Tougot 1
Centre d'Informatique
Universite de Technologie
60206 Compiegne, France

15. R. A. Langworthy 1
Applied Technology Laboratories
U.S. Army Research and Technology Laboratory
Fort Eustis, Virginia 23604

16. E. M. Lenoe 1
Army Materials & Mechanic Research Center
Arsenal Street
Watertown, Massachusetts 02172

17. Dr. Paris Genalis 1
Naval Ship Research and Development Center
Bethesda, Maryland 20084

18. C. Miller 1
Naval Sea Systems Command
Department of the Navy
Washington, D. C. 20362

19. Code SEC 6734 1
Naval Ship Engineering Center
Philadelphia Division
Philadelphia, Pennsylvania 19112

20. A. M. Diness (Code 471) 1
Department of the Navy
Office of Naval Research
Arlington, Virginia 22217

21. Ray M. Standahar — 1
Office of Secretary of Defense, DDR&E
3D1089 Pentagon
Washington, D. C. 20301

22. R. Rice (Code 6360) — 1
Naval Research Laboratory
Washington, D. C. 20375

23. E. Van Reuth — 1
Defense Advanced Research Projects Agency
1440 Wilson Boulevard
Arlington, Virginia 22209

24. I. Machlin — 1
Naval Air Systems Command
Department of the Navy
Washington, D. C. 20361

25. B. Probst, MS 49-3 — 1
NASA-Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

26. Mr. C. P. Blankenship, MS105-1 — 1
NASA-Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

27. Dr. H. Graham/AFML/LLM — 1
Department of the Air Force
Air Force Materials Laboratory
Wright-Patterson Air Force Base, Ohio 45433

28. Mr. George Strong — 1
DCASMA, Phoenix
3800 North Central Avenue
Phoenix, Arizona 85012

29. AFML/LLM/N. M. Geyer — 1
Air Force Materials Laboratory
Wright-Patterson Air Force Base, Ohio 45433

30. S. Wiederhorn — 1
National Bureau of Standards
Washington, D. C. 20234

31. Allan F. Greiner — 1
United Technologies Research Center
East Hartford, Connecticut 06108

Thesis
P9135
c.1

Preisel

175340

Stress analysis of ceramic gas turbine blades by the finite element method, part II.